ISSN 0868-6157

КОМПЬЮТЕР ПРЕСС

4'93

КОМПЬЮТЕР ПРЕСС

КОМПЬЮТЕРНАЯ ГРАФИКА

Увидеть невидимое 3

Искусство преображения информации 11

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

Jumbo из Колорадо 17

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Профессионалы, это для вас: библиотека Object Professional 21

РАЗГОВОРЫ

Назад в будущее, или Кое-что о классике предмета 29

ЗАЩИТА ПРОГРАММ И ДАННЫХ

Регистрация работы отладчика 37

НОВЫЕ ПРОДУКТЫ

Новая версия ALDUS PAGEMAKER — что нового? 41

ПЕРСОНАЛИИ

ELCO Technology 42

Computer Associates. Фейерверк программных продуктов 52

Мерисел: десант в Сибирь 54

TRIDENT уже пришел 57

ВПЕЧАТЛЕНИЯ

"Рабочие лошадки" от Epson 45

КАК ЭТО РАБОТАЕТ

Принтеры с "подогревом" 50

НАМ ПИШУТ

Об одной ошибке операций с плавающей точкой 59

ТЕНДЕНЦИИ

Инструментальные возможности открытой текстовой среды 65

СЕТИ

Локальная сеть и шахматы, или Почему вы не используете NetBIOS? 71

ИГРЫ

Шерлок Холмс против Джека Потрошителя 77

НОВОСТИ 78

4'93



КОМПЬЮТЕРПРЕСС

Издается с 1989 года
Выходит 12 раз в год
4'93 (40)

Обложка создана с помощью пакета научной графики epiGraf фирмой Bark. Дизайн П.М.Косткина

Адрес редакции:
113093 Москва, аб.ящик 37
Факс: (095) 200-22-89
Телефон для справок: (095) 471-32-63
E-mail: editorial@computerpress.msk.su

Реклама в номере:

Interlink	10
КАРАТ-2000	16
Пеленг	19
ИнФоС	20
STEEPLER	27
Алмаз-42	33
Мика СНТ	33
Лааль	33
ParaGraph	36
Сектор	40
Красная Волна	44
Symantec	44
Интерсофт	46
Совин, Novex	47
САЛМА	49
Микромакс	58
APS-COM	63
ServerSoft	64
RELCOM	67
АТД Интернэшнл	76
НПО "ВМИ"	76
ComputerPress shop	80

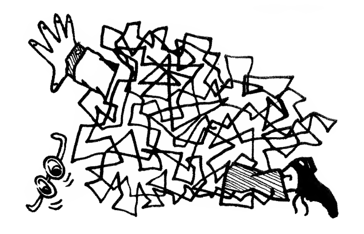
Ответственность за информацию, приведенную в рекламных материалах, несет рекламодатель.

Сдано в набор 28.02.93. Подписано к печати 10.03.93. Формат 84x108/16. Печать офсетная. Бумага типографская. Усл.печ.листов 8,4+0,42 (обложка). Кр.-отт. 10,08.Тираж 52000 экз. Заказ 3545. С-16.

Оригинал-макет подготовлен агентством «КомпьютерПресс».

Тексты проверены системой «ОРФО».

Отпечатано в полиграфической фирме «Красный пролетарий» РГИИЦ «Республика». 103473 Москва, И-473, Краснопролетарская, 16.



Очень интересная статья.

Увидеть невидимое

"Понятие визуализации в научных исследованиях включает совокупность средств и способов увидеть невидимое. Под "невидимым" понимается набор данных, описывающих какое-либо явление или процесс".

"В тот момент, когда требуется графическое отображение данных... пользователь попадает в среду графического пакета epiGraf, может посмотреть на свои данные и вернуться обратно в покинутую программу".

"Между данными и графической программой может находиться произвольно сложный алгоритм, который преобразует фактически имеющиеся данные в то, что следует изобразить".

"...с компьютером работает все большее число людей и большинство из них не имеют возможности и времени подробно изучать работу программ, а хотят быстро получить конечный результат. Поэтому программы должны быть такими, чтобы с ними можно было работать даже не читая описания или, по крайней мере, обращаться к описанию только в сложных ситуациях".

"...мы предусмотрели перенастраиваемую систему горячих клавиш, которые могут определяться пользователем, а также возможность запоминать небольшие макрокоманды".

"...имеется способ работы... Пользователь просто пишет последовательность команд и параметров в текстовом командном файле, запускает графическую программу пакета с указанием имени этого файла и идет пить кофе или спать. Так можно автоматически отобразить сотни наборов данных... Такой способ незаменим в том случае, когда надо поручить обработку данных человеку, который органически не воспринимает компьютер".

Введение

Авторы настоящей статьи — физики-теоретики. Принято считать, что теоретику для работы достаточно иметь неограниченное количество бумаги и что-нибудь пишущее. Иногда еще требуется калькулятор. Еще реже возникает необходимость отдать на счет на большой машине колоду перфокарт того или иного размера и получить через день многометровый рулон бумаги с колонками цифр. В бывшем СССР несколько лет назад перечисленные средства практически исчерпывали имеющиеся возможности.

В то же время формулы в современной науке становятся все более сложными и менее наглядными, а получаемые от большой машины колонки цифр также не обладают большой информативностью (если их больше, чем помещается на одной странице). Существовали, конечно, теоретические возможности получить на большой машине результаты в графическом виде, но на практике они были почти недоступны. Велика же была наша радость, когда у нас появились первые советские персоналки серии ДВК с графическим дисплеем и графическим принтером. Радость несколько омрачалась почти полным отсутствием программного обеспечения для машинной графики на этих компьютерах. Но примерно в течение года эта проблема была нами снята путем написания довольно большого количества графических программ для визуализации наших данных.

Прошло немного времени, появились персональные компьютеры серии IBM. Нам казалось, что тут-то можно будет воспользоваться готовым программным обеспечением, которое сможет делать то, что мы получали на ДВК, гораздо быстрее и лучше. Все оказалось по-другому. Конечно, для IBM PC написано довольно много графических пакетов. Но на поверку оказывается, что в большинстве случаев это не то, что требуется. Практически единственными полезными для визуализации и чаще всего используемыми физиками пакетами оказались довольно старые продукты GRAFOR и Surfer компании Golden Software. Ограниченные возможности, предоставляемые для визуализации имеющимся программным обеспечением, подтолкнули нас к собственной работе в этом направлении. Часть наших разработок реализована в коммерческом пакете epiGraf.

В настоящей статье мы попытались изложить соображения по поводу визуализации данных с точки зре-

ния физиков, но надеемся, что они будут интересны не только физикам, но и всем тем, кто имеет дело с необходимостью наглядного отображения данных.

Что такое визуализация

Понятие визуализации в научных исследованиях включает совокупность средств и способов увидеть невидимое. Под "невидимым" понимается набор данных, описывающих какое-либо явление или процесс. Основная задача визуализации — придать данным наглядность, поставив им в соответствие адекватное графическое изображение. В результате при анализе данных в работу включается ответственная за зрение часть человеческого мозга, значительно превосходящая по объему и возможностям ту часть, что отвечает за математические способности.

В докомпьютерную эпоху вся визуализация сводилась к возможности построить вручную на миллиметровке двумерный график какой-либо зависимости X(Y) (одномерные данные). А если требовалось построить что-либо типа карты или горного пейзажа (трехмерный график), то есть отобразить двумерные данные Z(X,Y), то это требовало воистину титанических усилий. Персональный компьютер позволяет в долю секунды построить одномерный график и за небольшое число секунд — трехмерный график (на графических станциях, которые пока доступны только достаточно богатым западным научным организациям, на это уходят тоже доли секунды).

На вопросах построения простейших двумерных графиков X(Y) на компьютерах мы не будем подробно останавливаться. Идейно процесс их построения достаточно прост, и основной проблемой, которая более или менее успешно решается в существующих пакетах, является организация удобного интерфейса с пользователем, особенно если графиков очень много. Основная часть статьи посвящена визуализации многомерных данных (много — это два или три). Но сначала речь пойдет об организации самих данных, подлежащих визуализации.



Типы данных и их организация

Мы будем говорить об отображении данных, которые имеют M компонент и которые можно естественно упорядочить с помощью N независимых параметров, и будем называть их M-компонентными N-мерными данными.

Приведем несколько примеров. Зависимость температуры воздуха в одной точке от времени суток — однокомпонентные одномерные данные (независимый параметр — время). Температура в разных местах пока еще Ленинградской области в один момент времени — однокомпонентные двумерные данные (независимые параметры — расстояние до Санкт-Петербурга и азимутальный угол шпиля Петропавловской крепости из точки наблюдения). А если мы измерим температуру на территории в зависимости от времени, то получим трехмерные однокомпонентные данные (третье измерение — время). Еще один метеорологический пример теперь уже многокомпонентных данных — данные по температуре, давлению, скорости ветра и влажности на какой-либо территории (четырехкомпонентные двумерные данные).

Перейдем к вопросу об универсальной организации данных, удобной для визуализации на компьютере. Первый приходящий в голову способ организации N-мерных M-компонентных данных в виде M штук N-мерных массивов, несмотря на простоту использования, нельзя назвать ни универсальным, ни удобным. Так, в приведенном примере с температурой в регионе точки измерения расположены не на равномерной прямоугольной сетке, а в местах, где удалось измерить. Возможен также вариант, когда имеются в наличии не непосредственно визуализуемые данные, а какие-либо их интегральные характеристики (например, фурье-коэффициенты).

Перебрав ряд возможностей, мы пришли к выводу, что наиболее оптимальной организацией данных будет алгоритм (программа), на вход которого подаются значения N независимых параметров и отображаемые данные, а на выходе получается M компонент данных, отвечающих входным параметрам. Конечно, идея неплохая, но MS-DOS, используемая на персональных компьютерах, не имеет встроенных средств для обмена данными между программами (помимо обмена через файлы). Но в то же время в MS-DOS возможно одновременное присутствие в памяти нескольких программ. Когда в оперативной памяти находятся две программы, они могут обмениваться информацией, передавая уп-



равление друг другу через программные прерывания в соответствии с определенными соглашениями.

Для реализации такого обмена мы зарезервировали три вектора прерываний. По одному вектору программа может узнать, имеется ли уже в памяти ее партнер и что он может делать (например, сообщается размерность и число компонент обрабатываемых данных). А два других вектора обслуживают две точки входа в программу, организующую интерфейс с данными. По одной точке входа программа, генерирующая изображение, требует у программы — драйвера данных вернуть значение данных в заданной точке в пространстве независимых параметров. По другой точке входа (так называемой инициализационной) управление просто передается программе-драйверу, который может сделать, что ему потребуется (например, прочитать что-нибудь с диска или инициализовать какие-нибудь массивы).

Описанное взаимодействие между графическим модулем и драйвером данных выглядит, возможно, сложновато для простых пользователей персональных компьютеров (не системных программистов). Но фактически процесс создания драйвера данных выглядит достаточно просто. На языке высокого уровня (например, FORTRAN, C, Pascal) пишется исполняемая программа, которая должна содержать помимо всего прочего две подпрограммы, обслуживающие две точки входа в драйвер данных, описанные выше. В тот момент, когда требуется графическое отображение данных, программа вызывает специальную написанную нами на ассемблере подпрограмму, передавая ей имена процедур обработки запросов графического модуля. Процедура на ассемблере инициализирует нужные векторы прерываний и загружает в память графический модуль с помощью библиотечной функции типа SPAWNL, которая есть в большинстве существующих на IBM PC компиляторах. После этого пользователь попадает в среду графического пакета epiGraf, может посмотреть на свои данные и вернуться обратно в покинутую программу.

Для пользователей, которые совсем не пишут программ, мы заготовили уже готовые драйверы, обслуживающие простейшие стандартные типы данных. А именно: многомерные массивы чисел, заданные в точках равномерной или неравномерной или случайной сетки и хранящиеся в файле, а также данные, заданные в виде несложных математических формул (например, F(x,y)=x*x-sin(y*y)). Эти драйверы могут быть загружены непосредственно из графических модулей. Помимо организации интерфейса между графическим модулем и данными стандартные драйверы позволяют в интерактивном режиме осуществлять основные операции с данными: просмотр, редактирование, изменение формата и т.д. После того как мы обсудили типы и организацию данных, перейдем к вопросу о том, как данные можно изображать.

Типы изображений

В докомпьютерную эру для визуализации многомерных данных ученые использовали практически один тип изображений: изображение двумерных данных в виде географической карты с изолиниями и цветной раскраской в зависимости от, например, высоты местности.

В раннюю компьютерную эпоху появилась возможность строить трехмерные графики функций двух переменных в виде так называемых каркасных или проволочных (wire frame) графиков. При этом практикуются изображения как с удалением невидимых линий, так и без оного. Упомянутый пакет Surfer изготавливает такие графики достаточно качественно и быстро. Несомненная положительная черта этих изображений — в возможности организации их вывода на любое внешнее устройство без потери точности изображения. В принципе таких графиков достаточно для отображения сравнительно гладких поверхностей. Но, несомненно, отрицательная черта таких графиков заключена в резком снижении наглядности изображения в случае, если данных много и они описывают не слишком гладкую поверхность. Поэтому большое значение имеет возможность изображать поверхность не схематически в виде набора линий, а с некоторыми элементами фотореалистичности с освещением источником света. Конечно, построение таких изображений требует несравнимо большей вычислительной работы и до недавнего времени было возможно лишь на графических станциях. Современные персональные компьютеры подошли к границе, когда и они позволяют строить близкие к фотореалистическим изображения за разумное время, измеряемое минутами. Эта возможность была реализована нами в пакете epiGraf.

В качестве модели освещения мы взяли один удаленный источник белого света, угловые размеры которого

могут варьироваться. Также могут варьироваться такие характеристики поверхности, как коэффициенты диффузного и зеркального отражения. Вообще говоря, для передачи плавного перехода полутонов необходим видеоадаптер с большим числом одновременно доступных цветов. Но если использовать смешивание на экране близких по интенсивности цветов с помощью метода, аналогичного используемому при печати на черно-белом принтере методу Байера, то можно получить приемлемые результаты даже на простейшем видеоадаптере EGA. Если же работать на VGA-адаптере, допускающем 64 уровня яркости для каждого основного цвета, то можно получать прекрасные полутоновые фотореалистические изображения с использованием всего 16 цветов (рис. 1).

Интересно отметить, что качество 16-цветного изображения с высоким разрешением 640x480 выше, чем качество 256-цветного полутонового изображения с низким разрешением 320x200.

Вернемся к тому, как можно использовать при визуализации цвет. Типичный пример географической карты всем знаком. Там оттенками синего цвета изображают глубину моря, а оттенками коричневого цвета — высоту гор, то есть цветом отображается Z-координата графика F(x,y). Очевидно, что с помощью цвета можно отображать и что угодно другое, например плотность населения или уровень осадков. Таким образом, при построении трехмерных графиков возникает необходимость в изображении на трехмерной поверхности значений какой-либо дополнительной компоненты данных. Ее значения можно отображать либо различной интенсивностью одного цветового тона, либо разными цветами с одной интенсивностью, либо линиями одинакового значения дополнительной компоненты (изолиниями), нанесенными на трехмерную поверхность. Подчеркнем, что в этом случае мы получаем возможность рассматривать одновременно две компоненты данных в виде высоты и цвета поверхности.

До сих пор речь шла только об отображении данных, которые можно представить в виде трехмерного графика Z=F(X,Y) над плоскостью (X,Y). Часто встречаются данные, область определения которых не разворачивается в плоскость. Например, данные, заданные на поверхности сферы или тора. Изображать такие данные, естественно, можно как график над поверхностью, на которой они заданы. То есть в каждой точке области определения проводится нормаль к ней, длиной, равной значению компоненты данных в этой точке, и через концы нормалей проводится поверхность (рис. 2).

Для визуализации такого сорта данных мы создали программу отображения параметрически заданных поверхностей. С помощью параметрически заданных поверхностей можно, конечно, изображать не только графики над неплоской областью определения. Перечислим основные возможные варианты.

1. Изображение трехмерной траектории какого-либо процесса в виде трехмерной изогнутой трубки. Причем на трубке можно отображать дополнительную информацию, например время, с помощью модуляции ее формы, толщины и цвета (рис. 3).

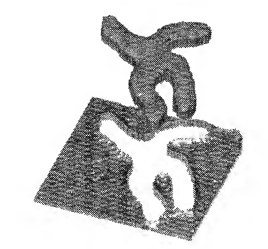


Рис. 1

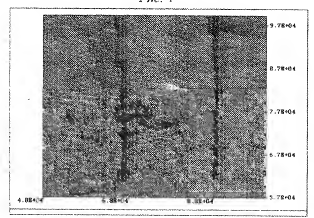


Рис. 2

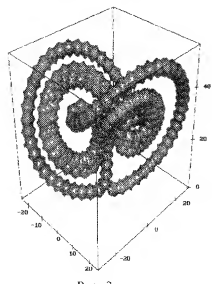


Рис. 3

2. Изображение области значений трехкомпонентных двухпараметрических данных в виде поверхности X=Fx(u,v), Y=Fy(u,v), Z=Fz(u,v), например компоненты векторного поля в каком-либо плоском сечении.

3. Изображение деформации начальной поверхности при развитии процесса во времени, например, в потоке жидкости. В этом случае для отображения процесса необходимо уже иметь несколько последовательных изображений (рис. 4).

Обсудим возможности отображения трехпараметрических данных, то есть данных, заданных в каждой точке объема, например плотность распределения каких-нибудь минералов внутри Земли или плотность распределения заряда в атоме или молекуле. Объем изображать значительно сложнее, так как мы и так уже проектируем трехмерный объект на плоский экран, а для отображения внутренних характеристик объекта нам просто не хватает степеней свободы плоского изображения. Приемлемые результаты по визуализации трехпараметрических данных можно получить, рассматривая их срезы при фиксированном значении одной из координат, например Z-координаты. Принципиально другая возможность — построить набор трехмерных изоповерхностей (поверхностей постоянного значения отображаемых данных) (рис. 5). Но совместить несколько изоповерхностей на одном изображении может оказаться затруднительным, так как, вероятно, большая часть изоповерхностей будет скрыта от наблюдателя одной, самой внешней поверхностью. Возможные выходы из затруднения:

- строить серию изображений изоповерхностей с разным значением отображаемой компоненты данных;
- строить несколько изоповерхностей, но прорезать в них сквозные отверстия там, где есть желание заглянуть внутрь;
- строить полупрозрачные изоповерхности, сквозь которые видно те элементы, которые они скрывают.

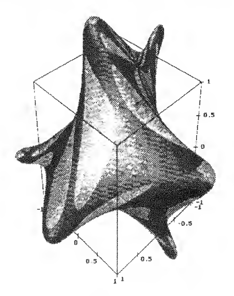


Рис. 4

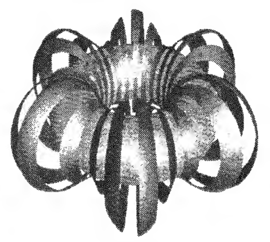
Последний вариант предъявляет повышенные требования к одновременно доступному числу цветов видеоадаптера (не менее 256) и реализован только в готовящейся к выпуску новой версии пакета epiGraf.

Еще один подход к отображению объемных данных, реализованный в нашем пакете: построение объемного рентгеновского снимка области, где заданы данные. Для моделирования рентгеновского снимка используется источник света, расположенный за объектом, и для каждой точки изображения вычисляется интенсивность прошедшего через объект луча. Причем считается, что коэффициент поглощения света в единичном объеме пропорционален значению отображаемой компоненты данных. Метод позволяет получать приемлемые результаты для сильно неоднородных данных.

Основные алгоритмы

Остановимся кратко на основных алгоритмах, которые используются в научной визуализации. Вопросы преобразования координат трехмерных объектов в двумерные координаты экрана подробно описаны в многочисленной литературе по машинной графике, и их мы затрагивать не будем. В конце концов создание изображения сводится к выводу на экран множества многоугольников с заданными экранными координатами; координаты в направлении, перпендикулярном экрану, отображаются значениями цветов и интенсивностей в вершинах. Возможно, через многоугольник могут еще проходить несколько семейств изолиний, которые также определяются значениями каких-либо величин в вершинах.

Стандартные, доступные на персональных компьютерах графические библиотеки предоставляют только возможности вывода одноцветных (или с заданной цветовой маской) многоугольников. Поэтому приходится создавать собственные процедуры. По сути дела, достаточен универсальный алгоритм, подобный методу закраски Гуро, но интерполирующий в каждую точку


Рис. 5

внутри многоугольника не только значения интенсивностей в вершинах, но и другие необходимые характеристики, такие как цвет или Z-координату.

Существенным моментом при выводе многоугольника является необходимость определить, будет ли он весь или его часть видна на окончательном изображении. Эта проблема носит название задачи удаления невидимых линий и поверхностей. Существует довольно много методов ее решения, из которых особой простотой программирования и использования отличается класс методов, ориентированных на растровую графику, в которых возможно определение видимости или невидимости многоугольника непосредственно в момент его вывода на экран и нет необходимости помнить все многоугольники или производить их предварительную сортировку. Эти алгоритмы можно реализовать на самом низком программном уровне (в современных графических станциях они реализуются аппаратно).

Перечислим эти методы.

1. Так называемый "алгоритм художника", который отличается тем, что вообще не анализируется видимость многоугольников. Просто многоугольники выводятся на экран все, но в должной последовательности: сначала дальние от наблюдателя, затем ближние. Такой алгоритм хорошо подходит, например, для изображения изоповерхностей, так как при изображении изоповерхности все равно надо сканировать по всей трехмерной области определения, чтобы не пропустить какой-либо ее фрагмент. Поэтому следует определить нужным образом порядок сканирования области, чтобы сначала проходить наиболее далекие от наблюдателя части.

2. Алгоритм, подобный методу плавающего горизонта, применяемому при построении трехмерных каркасных графиков функций двух переменных. Напомним кратко сам метод плавающего горизонта. Он предназначен именно для вывода каркасных графиков в виде горного пейзажа. Последовательно выводятся на экран фрагменты криволинейной сетки. Причем ближние к наблюдателю выводятся в первую очередь. При выводе очередного фрагмента проверяется, какая его часть видна выше или ниже области, определенной границами всех уже нарисованных фрагментов (верхняя и нижняя границы этой области и называются плавающими горизонтами). После вывода очередного фрагмента оба плавающих горизонта модифицируются добавлением к ним границ выведенного сегмента. Этот метод просто совмещается с алгоритмом поточечного проведения линии типа алгоритма Брезенхема. Для вывода закрашенных сегментов сетки, используемого при создании полутоновых изображений, алгоритм приходится немного модифицировать, чтобы вновь выводимые растровые сегменты не стирали уже нарисованных линий, а именно: пришлось ввести два типа плавающих горизонтов, один из которых контролирует видимость линий, другой — видимость точек из внутренних областей выводимых сегментов.

3. Наиболее универсальный из используемых алгоритмов — алгоритм с Z-буфером. Этот алгоритм позволяет выводить на растровый экран последовательность закрашенных многоугольников и линий в совершенно произвольном порядке. Идеологически алгоритм с Z-буфером предельно прост. Он использует тот факт, что экран монитора представляет собой совокупность отдельных точек, поэтому каждый выводимый на экран объект представляет собой не что иное, как множество точек. Если мы для каждой точки экрана будем запоминать Z-координату выводимых точек, а затем сравнивать Z-координату очередной выводимой точки с запомненной координатой и рисовать новую точку только если она ближе к наблюдателю, чем уже нарисованная, то мы решим тем самым проблему удаления невидимых линий и поверхностей. Современные графические станции используют для запоминания Z-координаты точек 3 байта, что требует для VGA-монитора с разрешением 640x480 пикселов около 900 Кбайт памяти. Поэтому мы ограничились глубиной Z-буфера 1 байт. Это обеспечивает 256 градаций глубины и вполне достаточно для большинства изображений. Кроме того, Z-буфер, занимающий 300 Кбайт, не хранится в памяти постоянно, а держится на диске и обновляется по мере необходимости.

После того как проблема удаления невидимых поверхностей решена, неожиданно возникает желание увидеть на экране, что же от нас скрыто (парадокс!). То есть появляется желание сделать объект слегка прозрачным (рис. 6).

Подчеркнем, что изображение прозрачного объекта коренным образом отличается от изображения объекта без удаления невидимых линий и поверхностей. Если мы просто нарисуем проволочный каркас, то еще можно как-то понять, что пытались изобразить, если же нарисовать как попало поверхность объекта, не заботясь об удалении невидимых поверхностей, то получится что-то абсолютно абстрактное. Самый простой алго-

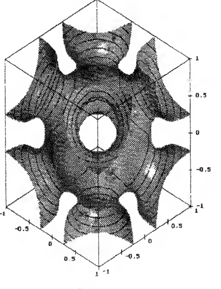


Рис. 6

ритм построения прозрачных объектов можно реализовать с использованием сочетания Z-буфера и аналогичного ему буфера интенсивностей прошедшего сквозь пикселы света. Программы, использующие алгоритмы построения прозрачных объектов, войдут в следующую версию пакета epiGraf.

Интерактивный анализ и обработка данных

При построении изображений в научной графике почти всегда упускают из виду, что изображение ценно не только само по себе, как картинка. Очень важно то, что в процессе созерцания изображения исследователя может осенить какая-то идея и ему захочется ткнуть пальцем (условно) в какую-либо точку на картинке и выяснить, а каким же цифрам соответствует данная точка. Нам неизвестны какие-либо графические пакеты (кроме epiGraf, конечно), предоставляющие такую возможность.

Мы реализовали в своем пакете набор возможностей интерактивного редактирования и анализа данных. К редактированию относятся возможности выбрать и изменить значения данных в отображаемых массивах (что позволяет осуществить не только пресловутую подгонку теории под эксперимент, но и наоборот) и ввод различных математических формул. В новой версии пакета появится возможность при отображении массивов данных производить задаваемые формулами преобразования координат и данных (например, переход из прямоугольных координат в полярные) (рис. 7). А пока пользователи должны писать для таких целей простенький алгоритм на FORTRAN, Pascal или C.

К анализу данных относятся возможности указать, выбрать на изображении точку и узнать численные значения координат этой точки, значения функции и ее частных производных в этой точке, а также возможность найти локальные экстремумы (максимумы и минимумы) методом градиентного спуска.

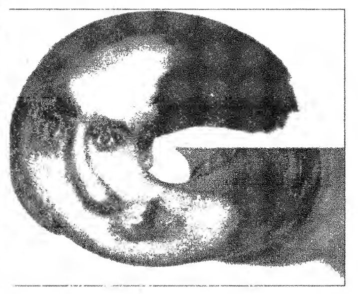


Рис. 7

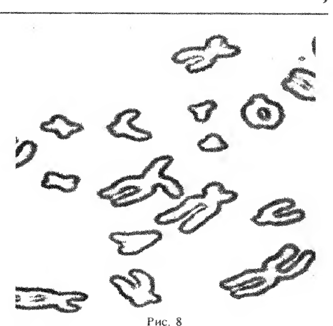


Рис. 8

Реализованный в пакете программный интерфейс графических модулей с отображаемыми данными предоставляет неограниченные возможности математической обработки данных. Между данными и графической программой может находиться произвольно сложный алгоритм, который преобразует фактически имеющиеся данные в то, что следует изобразить. Этот алгоритм может производить сглаживание зашумленных данных, выделять особо интересные области, фильтровать и т.д. Но, как известно, необъятное не объять, поэтому в рамках пакета мы реализовали только возможность отображения градиента функции в качестве дополнительной изображаемой компоненты данных (рис. 8). Все остальные способы обработки пользователи могут реализовать сами в виде программы.

Интерфейс

Мы достаточно поговорили об интерфейсе графического пакета с обрабатываемыми данными, вспомним и о пользователе, точнее об интерфейсе с ним.

К современному программному обеспечению для персональных компьютеров предъявляется жесткое требование его дружественности к потенциальному пользователю. Связано это прежде всего с тем, что с компьютером работает все большее число людей и большинство из них не имеют возможности и времени подробно изучать работу программ, а хотят быстро получить конечный результат. Поэтому программы должны быть такими, чтобы с ними можно было работать даже не читая описания или, по

крайней мере, обращаться к описанию только в сложных ситуациях. Программы должны в любой момент говорить пользователю, что он может в данный момент сделать и что он сделал неправильно.

Мы тоже отдали дань выполнению таких требований в виде моральных усилий и потраченного времени на создание обширной системы меню, поддержку мыши, контекстно чувствительную подсказку, обработку ошибочных ситуаций. Надеемся, что получилось нечто, с чем можно работать без предварительного обучения. Хотя, прямо скажем, в какой-то момент мы поняли, что управлять набором из примерно 200 параметров и функций заведомо не просто, и большая часть доступных возможностей на первом этапе работы с программами скрыта от пользователя и может быть включена специальным переключателем.

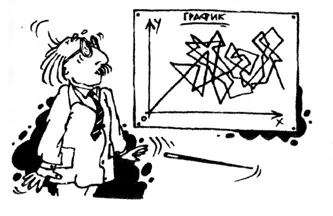
Но, поработав долгое время с интерактивной программой и достаточно изучив ее, пользователю надоедает "ползать" по многочисленным меню с целью изменения одного параметра. Для таких пользователей мы предусмотрели перенастраиваемую систему горячих клавиш, которые могут определяться пользователем, а также возможность запоминать небольшие макрокоманды.

А для случая, когда человек достаточно хорошо выучил, какие команды надо выполнить, чтобы построить требуемое изображение, имеется способ работы, напоминающий старый пакетный режим на больших машинах. Пользователь просто пишет последовательность команд и параметров в текстовом командном файле, запускает графическую программу пакета с указанием имени этого файла и идет пить кофе или спать (в зависимости от темперамента). Так можно автоматически отобразить сотни наборов данных, что в интерактивном режиме было бы затруднительно. Кроме того, такой способ незаменим в том случае, когда надо поручить обработку данных человеку, который органически не воспринимает компьютер. Опытный пользователь может подготовить несколько типовых командных файлов, написать короткую инструкцию, какие клавиши надо нажать, и смело ехать в командировку в надежде, что к его возвращению часть работы уже будет проделана.

Заключение

В небольшой статье невозможно описать все, касающееся визуализации в научных исследованиях. Невозможно даже описать все возможности, которые предоставляет наш пакет epiGraf. Мы постарались отразить наиболее полезные моменты, которые могли бы быть интересными для непрофессионалов в области компьютерной графики. И сам пакет рассчитан не на профессионалов-программистов, а на конечных пользователей.

В.Булатов,
В.Дмитриев



Искусство преображения информации

Современный персональный компьютер по своим возможностям лишь незначительно уступает профессиональным рабочим станциям, однако традиционно задачи машинной графики и обработки изображений считаются уделом рабочих станций и суперЭВМ. Сегодняшний день можно рассматривать как переломный момент, как прорыв в этом направлении, и тому есть веские причины.

1. Персональный компьютер сегодня доступен практически всем; он проник в самые маленькие лаборатории, учебные классы и не является редкостью даже на домашнем рабочем столе.

2. Развитые интерфейсные средства позволяют соединить персональный компьютер с любой экспериментальной установкой, с сетевыми базами данных и по телефонной линии даже с почтовыми и архивными серверами. Это привело к резкому увеличению потоков данных, обрабатываемых на персональном компьютере.

3. Большие успехи в разработке универсальных быстродействующих алгоритмов в области вычислительной геометрии и компьютерной графики ставят задачи графического моделирования и отображения в разряд рядовых для персонального компьютера.

4. И наконец, появление нового понятия — *визуализации* в корне изменило представления людей о предмете и задачах научной компьютерной графики (как это, впрочем, часто бывает — см. *Семодинамика*).

Сейчас мы все понимаем, что изображение — это лишь другой способ представления информации, преимущества которого становятся для человека особенно существенными, когда передаются большие объемы информации (такие, что позволяют сформировать зрительный образ объекта; для сравнения, 1 Мбайт — это таблица цифр на тысяче страниц или одно изображение 700x700 пикселов).

Зрительные образы в научном знании используются широко и зачастую являются единственной моделью объекта в нашем сознании: это всем известные графики функций одной переменной, гистограммы и более сложные — диаграммы нестационарной теории возмущений и т.п.

Однако всех их объединяет один недостаток — малый объем содержательной информации в изображении. Это можно понять, ведь каждое научное изображение требует большой технической и художественной работы. И вот наступает момент, когда большую часть этой рутинной и трудоемкой работы можно переложить на плечи персонального компьютера.

В настоящее время существует большое количество различных программ, ориентированных на научную и техническую графику, — от специализированных и встроенных в большие научно-технические комплексы и САПРы до программ общего назначения — но все они в той или иной степени страдают следующими недостатками (в скобках приводятся наиболее яркие представители).

1. Жесткие требования к формату и способу представления обрабатываемых данных (BOEINGgraph, MOLCAD).

2. Непродуманность или полное отсутствие возможности описания сложных областей определения данных (BOEINGgraph, GRAFOR).

3. Отсутствие возможности работы с многокомпонентными данными (SUPER, Mathematica, Matlab).

4. Отсутствие или предельная примитивность и недоработанность встроенных средств интерактивного редактирования данных (Surfer).

5. Полное отсутствие интерактивного анализа данных (99% всех программ, известных авторам).

6. Примитивный уровень работы с цветом, вплоть до полного отсутствия (Surfer, Mathematica, Matcad).

7. Недоработанные средства работы с палитрой или их отсутствие (Surfer, GRAFOR, ArtGRAF).

8. Недостаточно развитый интерфейс с пользователем и другими графическими системами.

Авторы настоящей статьи — физики-теоретики. Опыт освоения различных программных средств научной графики показал полную неудовлетворительность их технических, изобразительных и научных возможностей. Это вынудило нас создать свой комплекс программ TSTLIB 1.0/3.0 — epiGraf 1.0/2.1 (коммерческая версия) для ЭВМ серии ДВК и IBM PC.

Считаем необходимым поделиться своими соображениями и опытом в целях содействия скорейшему прогрессу в этой области. Надеемся, что статья сможет оказаться полезной профессионалам, любителям, тем, кто как и мы, отчаялся приобрести подходящую программу, и тем, кто еще только планирует приобретение программного обеспечения для научной и технической графики.

Статья разбита на параграфы, ориентированные на систематизацию, правильную постановку задачи и возможное решение в тех областях, где на сегодня обозначились серьезные недостатки или недоработки.

I. Цвет в научной графике

Трудно переоценить цвет в восприятии человека, большое количество людей вообще не признают черно-белых изображений. Чисто физиологически человек различает порядка $3.0x10^5$ цветов и оттенков; из них 256 градаций интенсивности, 200 градаций цветового тона и 10 уровней насыщенности. Таким образом, на цвет приходится 80% всей информации, содержащейся в изображении.

Долгое время в компьютерной графике цвет использовался только как средство классификации и выражения отношений порядка изображаемых объектов. Существенную роль здесь играло и то, что ранние средства отображения графической информации обладали крайне скромной цветовой палитрой.

Резкое удешевление оперативной памяти и появление цветных лазерных и струйных принтеров изменило ситуацию. Сейчас стало очевидно, что правильный выбор цвета даже для монохромных изображений очень сильно влияет на восприятие человека. Поэтому в современном программном обеспечении для компьютерной графики большое внимание уделяется средствам работы с цветом изображения. Здесь целесообразно выделить следующие аспекты.

1. Палитра

Большинство современных персональных компьютеров предоставляют средства для работы с палитрой — отображением информации видеопамяти в реальное цветовое пространство. Таким образом, уже после того как изображение построено (видеопамять заполнена информацией), имеется возможность изменять цвет этого изображения на мониторе.

Здесь сразу следует сделать оговорку: слово цвет в практике компьютерной графики используется в двух смыслах — собственно цвет, который формируется цифроаналоговым преобразователем видеоадаптера и который мы видим на мониторе, а также порядковый номер цвета — число, записанное в буферную память видеоадаптера.

Программное обеспечение предоставляет существенно различные средства для работы с палитрой. К сожалению, в большинстве случаев это средства самого низкого уровня для установки каждого отдельного цвета по его RGB- или CMY-координатам (соответственно интенсивности красной, зеленой и синей или голубой, пурпурной и желтой компоненты) в цветовом пространстве. Это средство заимствовано из ранних графических редакторов и, апеллируя к технической стороне этого вопроса, совершенно не учитывает особенности восприятия цвета человеком, которому ближе и нагляднее понятия *интенсивность*, *цветовой тон* и *насыщенность* (у художников — *разбелка*). Универсализация подходов к эталонным цветам и координатам в цветовом пространстве в основном выполнена в работах совещания МКО. Однако координаты МКО удобно использовать лишь для преобразований между различными стандартами, на практике же используются следующие координаты.

YIQ — телевизионный стандарт (NTSC)

- Y — у-координата МКО, интенсивность в базисе условных функций глаза;
- I — синфазный сигнал;
- Q — интегрированный сигнал.

Y хорошо описывает физиологическое ощущение интенсивности, однако две другие координаты выбраны из соображений оптимального размещения телевизионного сигнала в узком диапазоне частот и не очень удобны в задачах компьютерной графики.

YIQ- и RGB-координаты связаны линейным преобразованием

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{vmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \qquad (I.1.1)$$

HSV — координаты на шестигранном конусе

- H — цветовой тон ($0—360^0$);
- S — насыщенность (0—1);
- V — яркость (0—1).

```
V = max(R,G,B)
S = (V-min(R,G,B))/V
H = 60(1/(SV)( d(V-R)(G-B)+d(V-G)(B-R)+d(V-B)(R-G) )+
      (2+2d(V-B))d(V-R))
```

где

$$d(x) = \begin{pmatrix} 1 & x = 0 \\ 0 & x != 0 \end{pmatrix} \qquad (I.1.2)$$

TSL — координаты на трехгранном конусе

- T — цветовой тон (-1—1);
- S — насыщенность (0—1);
- L — уровень (0—1).

$$L=(R+G+B)/3$$
$$r=\sqrt{(R-L)^2+(G-L)^2+(B-L)^2}$$
$$S=r/L\sqrt{6}$$
$$T=sign(G-B)acos(2*(R-L)-(G-L)-(B-L))/\sqrt{6rPi} \qquad (I.1.3)$$

В последнем примере — формулы с большей вычислительной сложностью, но они позволяют получить лучший результат в задачах интерполяции цвета.

Выбор оптимальных координат в цветовом пространстве — это лишь половина дела. Всякие попытки подобрать палитру, используя лишь команды настройки отдельного цвета, приводят к плачевным результатам, поскольку нужно иметь большие художественные способности, хорошие глаза, хорошо настроенный монитор и много свободного времени.

Следует отметить, что большинство изображений содержат группы цветов с регулярно меняющимися параметрами. Самый простой пример — это монохромное изображение, тени и рефлексы на котором формируются цветами одного цветового тона и насыщенности с различной интенсивностью. С учетом этого обстоятельства многие программные продукты содержат средства для работы с контрастом и уровнем яркости всего изображения целиком. Это безусловно важно, но почти всегда необходимо работать лишь с частью всех цветов изображения, и, более того, это лишь частичное решение проблемы. Авторы считают, что для компьютерной графики совершенно незаменима возможность выполнять преобразования одновременно с произвольной группой цветов.

В собственной работе мы наиболее часто используем следующие групповые команды, реализованные в epiGraf.

1. **Выровнять координату R,G,B или T,S,L** — всей группе цветов присваивается среднее значение соответствующей координаты.

2. **Изменить координату R,G,B или, что более важно, T,S,L** — одновременно для всей группы производится сдвиг на константу соответствующей координаты.

3. **Упорядочить координату** — в выбранной группе значение координаты устанавливается в диапазоне от минимального до максимального значения в зависимости от номера цвета.

4. **"Изменить контраст"** — в выбранной группе цветов модифицируется расстояние от текущей координаты до среднего значения по группе. При этом вдоль выбранной координаты выполняется линейное растяжение в цветовом пространстве. Собственно изменение контраста является частным случаем такой деформации вдоль главной диагонали цветового куба (координата L).

5. **Интерполировать по группе** — установить в группе модифицируемую координату эквидистантно в диапазоне от значения координаты цвета с минимальным номером до значения координаты цвета с максимальным номером.

2. Модель освещения

В тех случаях, когда цвет передает реальные характеристики объекта (например, задачи компьютерного моделирования реальных сцен), он не может быть произвольно изменен. Круг таких задач в научной графике не очень широк, и цвет используется преимущественно для тонкого описания фактуры и мелких особенностей поверхности объекта. Здесь большую роль играет возможность подходящим образом выбрать параметры, описывающие физические свойства поверхности и характеристики освещения. Вообще, отражение и преломление света на поверхности — это сложный и многокомпонентный процесс, и количество необходимых для его описания параметров определяется главным образом постановкой задачи. Иными словами, необходимо определиться, что нужно от модели, а затем выбрать минимальное число интуитивно понятных параметров.

2.1. Назначение и характеристики освещения

В научной графике принято исходить из того, что освещение играет чисто вспомогательную роль и служит для подчеркивания и выявления особенностей поверхности или относительного расположения объектов. В этом случае собственные спектральные характеристики освещения несущественны и освещение предполагается монохромным или чисто белым.

Пространственное распределение источников света и их направленность оказывают очень большое влияние на качество изображения. Но в рассматриваемом случае не ставится задача полного описания освещения сцены, и целесообразно выделить три типа источников: это точечный удаленный источник, точечный близкий источник и источник рассеянного света, дающий общую фоновую засветку.

Вообще говоря, комбинациями этих источников можно достаточно полно смоделировать любое освещение, и фактически достаточно одного источника каждого типа. Во многих моделях освещения в научной графике не используется близкий точечный источник, так как пространственная неоднородность освещенности применяется, как правило, для видеоэффектов и приводит к уменьшению наглядности изображения в связи с неравноправностью элементов изображения.

Рассматриваемые примитивы освещения удобно характеризовать следующими параметрами: относительная интенсивность Iback, Ilamp, положение Xlamp, Ylamp, Zlamp и угловой размер точечных источников R. Последнее звучит странно, но нужно понимать, что понятие точечности означает лишь отсутствие интеграла по пространственному распределению, и угловой размер точечного источника вводится как характеристика размера зеркальных бликов на поверхности объектов.

2.2. Характеристики отражающей поверхности

Отражение света удобно рассматривать как совокупность двух процессов.

1. **Диффузное отражение** — яркость фрагмента поверхности пропорциональна его освещенности, которая в свою очередь определяется вкладами от источников

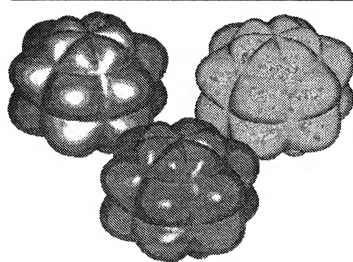


Рис. 1. Пример различных моделей отражающей поверхности

рассеянного света и точечных источников по закону Ламберта:

```
I = D*Iback + D*Ilamp cos(alfa)                    (1.2.1)
```

где D — коэффициент диффузного отражения, alfa — угол между нормалью к поверхности и направлением на источник. Для близкого источника второе слагаемое необходимо разделить на функцию расстояния до источника. Естественно предположить, что это квадрат расстояния, но на практике лучший результат получается при использовании линейной функции расстояния.

2. Зеркальное отражение — яркость фрагмента определяется углом между направлением на наблюдателя и вектором отражения teta:

```
I = M*Ilamp*f(teta,R)                              (1.2.2)
```

где M — коэффициент зеркального отражения (в более полных моделях освещения вводят спектральную и угловую функции отражения), а f — функция, описывающая угловую характеристику источника. Часто функцию f аппроксимируют степенной функцией

$$f=\cos(teta)^{R} \qquad (1.2.3)$$

2.3. Характеристики прозрачной поверхности

В современной научной визуализации все чаще используются прозрачные поверхности, несмотря на то что их построение требует привлечения более трудоемких вычислений и алгоритмов. Чаще всего применяется самая простая модель с показателем преломления, равным единице. Этого вполне достаточно для визуализации объемных распределений, и не требуется использования алгоритмов трассировки лучей с очень большой вычислительной трудоемкостью.

Интенсивность отраженного луча описывается формулами (I.2.1) и (I.2.2).

Интенсивность преломленного луча определяется следующим образом:

```
I = T*I0
T = (1-(4*Pi*D+M))*exp(-S*l))                      (1.2.4)
```

где T — коэффициент прозрачности, D и M — коэффициенты диффузного и зеркального отражения, S — коэффициент поглощения и l — путь луча в рассматриваемом слое. Если допустить, что все коэффициенты и путь луча не зависят от угла падения, то коэффициент прозрачности T можно считать константой. Интенсивность I0 — это интенсивность луча без рассматриваемого прозрачного слоя, описывающая вклад всех источников и отражающих поверхностей за ним.

На этом описание модели освещения можно закончить. Во многих задачах научной графики можно с успехом обойтись этой простой моделью, но следует сразу обозначить границы ее применимости — это цветные непрозрачные объекты, освещенные белым источником света, и монохромные или квазимонохромные (совокупность монохромных) прозрачные объекты с показателем преломления, равным единице. Для выхода за рамки этих ограничений необходимо использовать более полную модель освещения, которая учитывает спектральные и угловые характеристики всех коэффициентов.

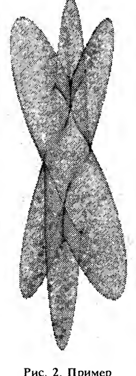


Рис. 2. Пример прозрачной поверхности

3. Использование цвета

Новый термин *визуализация* позволяет более полно осознать назначение цвета — это создание адекватного зрительного образа плюс передача информации.

Начнем с изображений, которые, по существу, являются копией реально существующего объекта. Все мы в детстве увлекались черно-белой фотографией и знаем, что даже фактура и цвет подложки влияют на наше восприятие. Используя бромпортрет или йодоконт, можно получать изумительные по выразительности фотографии, и наоборот, переусердствовав с виражем, можно получить отпечаток, который и в руки взять неприятно. Человек — он и в науке человек, и даже технические монохромные изображения



Рис. 3

обладают разной выразительной силой в зависимости от эмоционального соответствия объекта и цвета изображения. Это очевидное на первый взгляд соображение приводит порой к неожиданным результатам. Например, совершенно не очевидно, что мелкие особенности освещения и фактуры поверхности лучше видны при использовании малонасыщенной металлической палитры, и наоборот, пространственные отношения объектов нагляднее при использовании теплых тонов (рис. 3).

Не менее важен для восприятия контраст изображения и характер бликов на изображении (рис. 4), но если первое целиком определяется выбором палитры, то последнее зависит от выбора модели освещения и характеристик поверхности. Для мелких деталей важны доля зеркального отражения и угловые размеры источника. Общий контраст изображения определяется также долей рассеянного света.

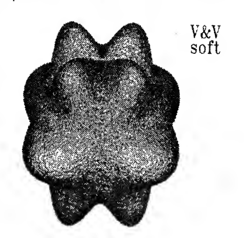


Рис. 4

Положение источника света — это очень тонкий вопрос. Проще всего обратиться к учебникам по фотографии. Можно лишь сказать, что наиболее часто используемая позиция источника — в верхней части сцены со стороны наблюдателя.

До сих пор речь шла о вспомогательном назначении цвета — выделении и подаче той информации, которая уже содержится в форме поверхности и расположении объектов. Но в научной графике цвет может играть и совершенно другую роль. Ограниченность возможностей рисунка по передаче информации определяется на сегодняшний день его принципиальной двумерностью, в то же время визуализуемые данные могут иметь большее число независимых компонент или принципиально большую размерность области определения; в том и в другом случае есть необходимость в одном изображении передать большее количество информации.

Достаточно привычным способом решения этой проблемы является построение псевдотрехмерных изображений (график функции двух переменных f(x,y)), являющихся в большинстве случаев проекцией представляемого трехмерного объекта. Недостающая размерность на психофизическом уровне восстанавливается по освещению и теням. Следующий шаг в использовании цвета — раскрашивание изображаемой поверхности цветом, зависящим от еще одной компоненты данных. Этот метод позволяет существенно продвинуться в изображении таких объектов, как физические поля, и с помощью одного изображения описывать корреляции независимых компонент многомерных данных.

Таким образом, в научной графике цвет — это, по существу, дополнительная размерность изображения. Принципиально — это три дополнительные размерности, в связи с трехмерностью цветового пространства. Однако в большинстве работ цвет симулирует только две размерности, так как в цветовом пространстве выбираются направления с максимальным разрешением — интенсивность и цветовой тон. Многомерные проекции не получили широкого распространения, поскольку у рядового человека они не вызывают никакого чувственного аналога.

II. Объект и способ его задания

Это специальный параграф для тех, кто занимается созданием собственной или универсальной графической системы.

В технической графике принято описывать объекты в виде совокупности примитивов, и вся работа проводится в пространстве сцены или в собственно экранных координатах.

В научной графике очень часто используются данные, заданные либо аналитически, либо численно в регулярной или случайной многопараметрической системе узлов. При этом данные могут иметь одну или несколько компонент (иными словами, несколько независимых значений в каждой точке области задания или узле сетки). Достаточно трудно найти общий подход к

представлению и редактированию данных, но можно выделить существенно общий момент — это размерность области определения и размерность области значений.

Программист, увлекающийся объектно-ориентированным программированием, хлопнет себя ладонью по лбу и скажет: "Да это же объект, получающий сообщение из области определения и возвращающий сообщение о значении данных". Действительно, с точки зрения программистов собственно процедуре построения изображения глубоко безразлична внутренняя структура данных и важно лишь количество передаваемых и возвращаемых значений. Дальнейший ход рассуждений достаточно очевиден — программу обслуживания данных можно выделить в отдельную единицу и менять по мере необходимости смены типа обрабатываемых данных. Именно такой подход реализован авторами при создании универсальных пакетов графических программ TSTLIB и epiGraf.

Программы обслуживания данных различного типа представляют собой обычные самостоятельные загрузочные модули и обмениваются сообщениями с графической программой по прерываниям или через фиксированные точки входа.

Достаточно очевидно, что даже при таком подходе невозможно охватить все варианты и схемы представления данных, но частично эта проблема решена путем создания открытой в этом направлении системы. Пользователь с минимальной квалификацией может написать собственную программу обслуживания данных по специальному шаблону, содержащемуся в пакете и реализующему стандартный алгоритм обмена данными.

Программа пользователя загружается из ведущей программы или сама загружает графическую программу. На этапе построения графическая программа выставляет запрос, содержащий массив с координатами точки и массив с дополнительными параметрами, а в ответ получает массив со значениями данных в этой точке.

Вся работа по вычислению или получению данных выполняется программой обслуживания; для редакции и модификации данных имеется дополнительная точка входа в программу обслуживания. Такое разделение обязанностей между графической программой и программой обслуживания данных позволяет разрабатывать наиболее удобный интерфейс для каждого специального типа данных. Более того, появляется возможность динамического получения данных прямо с экспериментальной установки или по удаленной линии, исчезают ограничения на формат и способ представления данных.

Как видно, интерфейс достаточно прост, универсален и может быть реализован в самых различных вариантах. Мы надеемся, что эти соображения смогут оказаться полезными при разработке универсальных интерфейсов с различными типами данных.

В.Булатов, В.Дмитриев

(Окончание следует)



Мы уже писали о том, что среди современных средств резервного копирования накопители на магнитной ленте по-прежнему занимают достаточно значительное место (КомпьютерПресс № 6'92, с.43-48). Для подавляющего большинства пользователей IBM-совместимых персональных компьютеров наибольший интерес представляют стримеры, отвечающие стандартам QIC-40/80. В этой небольшой статье мы расскажем о двух моделях подобных устройств — Jumbo 120 и Jumbo 250, которые выпускает фирма Colorado Memory Systems.

Jumbo из Колорадо

Усложнять — просто, упрощать — сложно.
(Закон Майера)

С появлением современных малогабаритных и, вообще говоря, достаточно надежных винчестеров для массы "неорганизованных" пользователей средства резервного копирования (и в частности стримеры) как бы отодвинулись на второй план. Разумеется, понятно, что при наличии локальной сети регулярное архивирование информации просто необходимо. И все же можно указать не менее 7 весомых причин (как это сделала, например, фирма Colorado Memory Systems), по которым и "неорганизованному" пользователю отказываться от приобретения стримера просто неразумно. Попробуем коротко прокомментировать эти причины.

От ошибок, разумеется, никто не застрахован. Как говорила подруга детства Великого Комбинатора Инга Зайонц, "и на старуху бывает проруха". Известно, например, немало случаев, когда вместо того, чтобы отформатировать дискету, пользователь начинал форматировать винчестер и спохватывался только тогда, когда уже была потеряна самая свежая или наиболее ценная информация. В этом случае именно стример мог бы оказаться своеобразным страховым полисом. Кстати, упомянутый ранее О.Бендер с большим уважением отзывался об этой бумаге, которая дает якобы полную гарантию.

Конечно, современные винчестеры практически не имеют сбоев в работе. Но много ли у нас в стране пользователей, которые имеют возможность обновлять свою "материальную часть" достаточно регулярно? Ну а "стареющий" винчестер, хотите вы этого или нет, постепенно превращается в "бомбу" замедленного действия.

Пожалуй, не секрет, что "винт" рядового пользователя, как правило, забит почти под завязку. Однако из всей хранимой, мягко говоря, информации используется интенсивно, дай бог, 30—40%, все остальное лежит обычно мертвым грузом: и стереть жалко, и переписать некуда. Нетрудно догадаться, что и в этом случае наличие стримера могло бы существенно облегчить жизнь, позволив освободить на жестком диске достаточно места.

На случай землетрясений, пожаров и других форс-мажорных обстоятельств иметь на кассете стримера копию всей информации со своего винчестера (или хотя бы ее часть) не так и плохо. Ну, скажете вы, парень загнул, еще бы про летающие "тарелки" вспомнил. "Тарелки", конечно, здесь ни при чем, а вот "спереть"

Параметры	Емкость картриджа Мбайт			Производительность, Мбайт/мин	
Модели стримеров	DC2000	DC2080	DC2120	"флоппи"	TC-15
Jumbo 120	40/80*	40/80	60/120	2,2	2,2/4,4*
Jumbo 250	–	83/166	125/250	2,2	4,4/8,8*

* – со сжатием данных/без сжатия

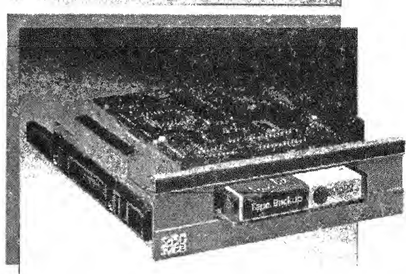
ваш компьютер (вместе, заметьте, с винчестером) в наши беспечальные времена могут очень даже запросто.

Программным обеспечением, занимающим несколько мегабайт, сегодня никого не удивишь. "Крутой софт" вместе с различными библиотеками, кстати, может занимать существенно больше места. Особенно "прожорливыми" до дискового пространства являются, в частности, различные системы проектирования (CAD/CAM), компьютерная графика. В этом случае, например при работе с заказчиком из другого города (и наличии, разумеется, у обеих организаций совместимых стримеров), передача большого объема информации на кассетах представляется очень удобной.

Не обходят обычно стороной "неорганизованного" пользователя и проблемы электропитания. Как правило, приобретение бесперебойных источников питания (UPS) затягивается, то ли из-за отсутствия денег, то ли от недопонимания проблемы. Может быть, в некоторых случаях покупку стримера обосновать будет легче?

Ну и еще один простой пример. Всем известно, что дефрагментация файлов на винчестере существенно замедляет доступ к записанным на нем данным (по некоторым сведениям в 2, а то и в 3 раза). Конечно, можно использовать специальные программы, которые позволяют переписывать имеющиеся на диске файлы последовательно кластер за кластером. Однако не следует забывать, что после восстановления информации со стримера в режиме file-by-file эта проблема также решается, но уже как бы автоматически.





Теперь непростой вопрос: какой же стример приобретать? Впрочем, ответ на него мы уже давали (см. КомпьютерПресс № 6'92, с.48). Стример для небольших локальных сетей, а также для "неорганизованного" пользователя должен отвечать стандартам QIC-40/80. Коротко напомним, почему.

Во-первых, это гарантирует хоть какую-то совместимость. К тому же это качество в данном случае обеспечивается не только в пределах одной спецификации, но и предполагает совместимость устройств "снизу-вверх". Таким образом, картридж (кассета), записанный на стримере, соответствующем стандарту QIC-40, будет читаться на устройстве, выполненном по стандарту QIC-80. Во-вторых, при подключении к компьютеру стримеров, отвечающих спецификациям QIC-40/80, как правило, не возникает особых трудностей и проблем. Такие устройства не зря называют "floppy tape" — дело в том, что они могут подключаться к существующему в любой "персоналке" контроллеру флоппи-дисков. Преимущества используемого технического решения часто просто очевидны. Тем не менее, фирмы-производители предусматривают и специальные контроллеры, которые позволяют увеличить скорость обмена данны-

ми, впрочем, о них речь еще впереди. В-третьих, на каждом картридже рассматриваемых устройств может сохраняться от 120 до 250 Мбайт информации (если использовать сжатие данных), что, как правило, превосходит или соответствует объемам имеющихся винчестеров. Немаловажным фактором является и тот, что стримеры, отвечающие стандартам QIC-40/80, выпускаются целым рядом крупных фирм, а это в свою очередь еще больше расширяет круг совместимых устройств.

Сегодня мы расскажем о двух моделях стримеров, которые, по мнению редакции КомпьютерПресс, являются лучшим выбором для использования их с отдельным компьютером или в небольшой локальной сети. Впрочем, это не только наше мнение. Модели стримеров Jumbo 120 и Jumbo 250 американской фирмы Colorado Memory Systems хорошо известны во всем мире. В их отношении ведущие компьютерные издания не скупятся на эпитеты: "Best Buy", "Perfect 10", "Editors' Choice" и т.п. Попробуем пояснить, почему.

Надо отдать должное специалистам фирмы Colorado Memory Systems, которые провели просто огромную работу, детально проверяя совместимость своих новых моделей стримеров с компьютерами разных фирм. Судя по предоставленной нам информации, было исследовано более 400 различных моделей компьютеров. Среди фирм-производителей, которых насчитывалось около 100, были, например, широко известные IBM, Dell, Hewlett-Packard, Acer, Compaq, DEC, NCR, NEC, Wyse, Zenith, Zeos, впрочем, всех, даже из числа известных, перечислить довольно сложно. Кстати, результатом этой работы были не просто ответы "годен-негоден". Для различных моделей компьютеров фирма Colorado Memory Systems предлагает теперь наборы специальных внутренних креплений и/или кабелей конкретно под свои стримеры. Например, CK-35 — для компьютеров HP Vectra или CK-60 — для изделий с маркой Epson.

Модели стримеров Jumbo 120 и Jumbo 250 отвечают спецификациям QIC-40 и QIC-80 соответственно. Пользователю предлагаются два варианта исполнения подобных устройств: встраиваемый и внешний. В последнем случае встраиваемое устройство "одевается" в изящный пластмассовый корпус, который вместе с кабелями и прочими аксессуарами входит в фирменный набор KE-10. Кстати, модель Jumbo 120 может использовать две скорости передачи информации — 250 и 500 Кбайт/с, что, как известно, совпадает со стандартными значениями для контроллеров флоппи-дисков компьютеров типа XT и AT. Стримеры Jumbo имеют программную поддержку для сжатия данных, причем фирменный алгоритм соответствует стандарту QIC-122. По заявлению фирмы Colorado Memory Systems, использование эффективного кодирования информации позволяет сжимать данные в отношении от 6:1 до 1,3:1 при среднем значении 2:1. Как известно, в основу алгоритма по QIC-122 был положен мощный алгоритм сжатия Лемпеля-Зива, который одной из первых эффективно начала использовать фирма Stac Electronic, разработав специальную микросхему кодера-декодера данных. Кстати, использование специального контроллера TC-15 (TC-15M для PS/2) вместо стандартного для флоппи-дисков позволяет увеличить производительность (скорость архивирования информации) моделей стримеров Jumbo 120 и Jumbo 250 с 2,2 до 4,4 и 8,8 Мбайт/мин соответственно (см. таблицу). Поскольку алгоритм сжатия в данном случае реализуется аппаратно, то на его выполнение необходимо практически "нулевое" время. Для контроля передаваемых данных используется циклический избыточный код (CRC), а для коррекции ошибок (ECC) — алгоритм Рида-Соломона.

Особо хотелось бы отметить и то обстоятельство, что контроллеры фирмы Colorado Memory Systems не требуют ручной установки переключателей и сведений о свободных адресах, номерах прерываний IRQ и каналах прямого доступа DMA. Конфликта между аппаратными средствами не возникает, поскольку используемое при инсталляции программное обеспечение применяет особый алгоритм тестирования, который в 95% случаев сам определяет необходимые параметры для установки. В оставшихся же 5% случаев необходимые установки производятся пользователем "программными" переключателями, которые отображаются на дисплее компьютера.

Программное обеспечение, поставляемое с моделями стримеров Jumbo, полностью совместимо с сетевыми пакетами таких фирм, как Novell, IBM, 3COM. Не следует также забывать, что для устройств, соответствующих спецификациям QIC-40/80, возможно использование, скажем так, общего программного обеспечения, например пакета PC Tools фирмы Central Point Software.

Jumbo — одни из самых недорогих и быстрых стримеров, предлагаемых на отечественном рынке. Это хороший выбор именно для вас!

А.Борзенко

ИНТЕГРАТОР "ВИКТОРИЯ"

Фирма ИнФоС представляет официальных дилеров

РОССИЯ

Москва
СП "Аквариус Системз Интеграл"
Телефон: (095) 906-00-88

НПМП "Инвюркомп"
Телефон: (095) 261-96-69

Новгород, Колмово
Фирма "Мединфо"
Телефон: (816-22) 2-37-88, 2-51-87
Факс: (816-22) 2-52-62

Новочеркасск, Ростовская обл.
Фирма "Иннотех"
Телефон: (863-52) 2-75-19, 2-30-84
Факс: (863-52) 2-33-24

МП "Инис"
Телефон: (863-52) 2-76-68, 2-46-59
Факс: (863-52) 3-04-42

Ковров, Владимирская обл.
Фирма "Синпро"
Телефон: (092-32) 3-21-22

Норильск
МП "Аквилон"
Телефон: (391-52) 4-00-10

Лысьва, Пермская обл.
Диагностический Центр
Телефон: (342-49) 2-07-87

УКРАИНА

Павлоград, Днепропетровская обл.
Многоотраслевая фирма "Градиент"
Телефон: (056-72) 6-48-31

Снежное, Донецкая обл.
ИВЦ "Торезантрацит"
Телефон: (256-22) 5-41-80

Севастополь, Крым
Севастопольский Деловой Дом
Телефон: (0690) 52-92-83, 57-59-72

БЕЛАРУСЬ

Минск
МГП "Норокон"
Телефон: (0172) 30-64-19

КАЗАХСТАН

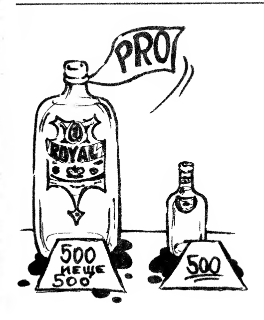
Кентау, Шимкентская обл.
Фирма "Жорга"
Телефон: (325-36) 3-39-91

УЗБЕКИСТАН

Ташкент
Фирма "Информпроспект"
Телефон: (3712) 45-85-93, 56-57-97

ПРИГЛАШАЕМ ДИЛЕРОВ

Телефоны для справок:
(095) 471-32-63, (084-39) 2-24-82
Адрес: 113093 Москва, а/я 37



Библиотека Object Professional (OPro) фирмы Turbo Power Software представляет собой наиболее полный (из имеющихся для компилятора Turbo Pascal) набор объектов (более 100), предназначенных для реализации пользовательских интерфейсов, включая меню, списки, средства просмотра и редактирования файлов, панели и поля ввода информации, справочную систему, строчные и полноэкранные редакторы, а также большое число модулей для взаимодействия с операционной системой, управления памятью (дополнительной и расширенной), управления мышью, поддержки принтера (через порты, DOS или BIOS), средства для создания резидентных программ, управления прерываниями и многое другое.

Профессионалы, это для вас: библиотека Object Professional

В этом обзоре рассматривается версия 1.20 библиотеки OPro, любезно предоставленная нам фирмой Turbo Power Software. Данная версия поддерживает компиляторы Borland Pascal 7.0/Turbo Pascal 7.0 и была выпущена одновременно с этими компиляторами (ноябрь 1992 года).

Взгляд сверху: основы организации библиотеки

Если кратко охарактеризовать библиотеку OPro, то ее можно разделить на три составляющие: управление экраном, ввод/вывод данных и системные модули. Документация (три тома) представляет библиотеку именно таким образом. В каждом разделе, помимо объектов высокого уровня, существуют вспомогательные объекты, а также процедуры и функции, большинство из которых используются непосредственно основными объектами.

Управление экраном

В этом разделе библиотеки мы можем найти все, что необходимо для продуктивного и профессионального управления экраном: от модуля OPCRT, являющегося расширением функций модуля CRT стандартной библиотеки (поставляемой в составе компилятора), до комплексных модулей OPWINDOW и OPFRAME, объединяющих в себе объекты для реализации окон, рамок и меню.

К модулям управления экраном относятся также OPMOUSE — набор подпрограмм поддержки мыши и OPCMD — предоставляющий механизм трансляции нажатых клавиш и кнопок мыши в команды и вклю-

чающий два объекта — CommandProcessor и CommandPacker. Функционально объект CommandProcesor можно сравнить с методом GetEvent в Turbo Vision-приложениях.

К модулям управления экраном также можно отнести и модуль OPHELP, использование которого позволяет реализовать справочную систему для создаваемого приложения.

Ввод/вывод данных

Большинство объектов этой категории являются наследниками объекта CommandWindow. Сюда включаются объекты для реализации мемо-полей, текстовый редактор, по функциональности не уступающий редактору среды разработчика (ИСР компилятора), множество объектов, реализующих различные поля ввода с возможностью просмотра, редактирования и проверки введенных данных, а также модули для вывода информации на устройство печати.

Системные модули

Основным объектом, включенным в эту категорию, является объект OPROOT — наследник большинства объектов в иерархии OPro. Кроме того, реализованы "классические" объекты — стек, очередь, простой и двойной связанный список, а также множество других объектов низкого уровня, используемых другими объектами: BitSet — эквивалент наборов Turbo Pascal, но вмещающий до 65534 элементов, LargeBitSet — набор, включающий до 8388698 элементов, словарь StringDict, который может использоваться при написании парсеров, интерпретаторов и подобных утилит, объект StringArray для хранения строк и модуль OPLARRAY, содержащий объекты для хранения больших объемов данных в оперативной памяти, EMS или на диске. В последнем случае объем массива может достигать 32 Мбайт. При использовании программ, работающих в защищенном режиме (PMode-программ), с помощью модуля OPLARRAY возможно хранение до 16 Мбайт информации в оперативной памяти.

В OPro поддерживаются также потоки, в которых могут храниться объекты. Как и в Turbo Vision, объект должен быть зарегистрирован перед использованием в потоке, но потоки OPro и Turbo Vision не совместимы по формату. Объект Library является эквивалентом ресурсов в Turbo Vision.

Интересной является возможность создания резидентных программ, ядро которых занимает всего 6-8 Кбайт. Следование простым правилам позволяет преобразовать практически любую программу в резидентную, вызываемую командной клавишей.

К категории системных модулей можно отнести также модули поддержки работы с расширенной и дополнительной памятью и модуль OPXMS для работы с сегментами "верхней" памяти.

Помимо основных модулей, целая дискета в комплекте поставки содержит объекты и утилиты, объединенные термином "BONUS". Сюда включаются различные расширения для OPro, созданные как самой фирмой, так и многочисленными пользователями.

После того как вы получили предварительное представление о библиотеке OPro, давайте рассмотрим ряд основных модулей более подробно.

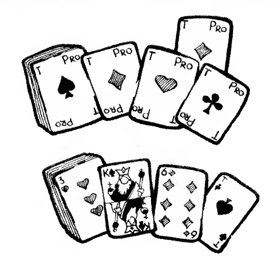
Модуль OPCRT

Как мы отметили выше, модуль OPCRT полностью заменяет функции модуля CRT стандартной библиотеки. Вдобавок к этому OPCRT содержит большое число функций для отображения информации. Отметим, что в отличие от функции Window стандартной библиотеки, в OPCRT имеется возможность сохранения области экрана, закрываемой окном и ее восстановления после удаления окна. Функции модуля OPCRT поддерживают работу с расширенной клавиатурой: имеется возможность проверки типа клавиатуры и обработки "расширенных" клавиш — F11 и F12. Для работы с расширенной клавиатурой имеется специальный модуль — OPENHKBD.

Две процедуры из этого модуля — StuffKey и StuffString могут быть полезны при передаче параметров дочерним процессам (Exec модуля DOS или ExecDos модуля OPDOS) и создании демонстрационных программ.

Использование цветов

В отличие от используемых в Turbo Vision палитр, позволяющих задавать цвета компонентам отображаемых объектов, в OPro существует специальный объект — ColorSet. Этот объект содержит 31 пару видеоатрибутов, описывающих цвета всех элементов отображаемых объектов, реализованных в OPro. Помимо



того, ColorSet содержит методы для изменения значений отдельных полей. Например, поле, задающее цвет вывода информации внутри окна, называется TextColor, а метод для изменения значения этого поля называется SetTextAttr.

При инициализации объекта указывается набор цветов, который используется для отображения этого объекта:

```
Memo.InitCustom(2, 4, 79, 24,
                MemoColorSet,{Набор цветов для объекта Memo}
                DefWindowOptions OR wBordered,
                BufSize,
                @Buffer);
```

Отметим, что в отличие от палитр в Turbo Vision, где реальный цвет отображаемого объекта зависит от его владельца (цвет определяется смещением в палитре владельца): поместите объект типа TInputLine в объект типа TWindow вместо TDialog — и вы увидите, что произойдет, в OPro мы имеем дело с реальными цветами, что существенно упрощает управление ими.

Также имеется объект LoadableColorSet, позволяющий использовать наборы цветов в потоках.

Для программистов, которым проще использовать "описательные" значения цветов (типа ЗеленыйНаКрасном) вместо цифровых значений, поставляется модуль OPCOLOR. С его помощью, например, атрибуту $3F придается в соответствие константа WhiteOnCyan.

Поддержка мыши

В отличие от большинства существующих реализаций модулей поддержки работы с мышью, в OPro проводится аналогия между функциями поддержки курсора (модуль OPCRT) и поддержки мыши (модуль OPMOUSE). Так, например, функции MousePressed соответствует функция OPCRT.KeyPressed, функции MouseGotoXY — функция OPCRT.GotoXY и т.д. Поддержка мыши указывается установкой соответствующей опции командного процессора для данного объекта:

```
XxxCommands.cpOptionsOn(cpEnableMouse)
```

Вызов подпрограммы EnableEventHandling активирует подпрограммы MousePressed и MouseKeyWord, которые функционально аналогичны подпрограммам KeyPressed и ReadKeyWord. MouseKeyWord возвращает псевдоскэн-коды, указывающие, какая кнопка мыши нажата.

Механизм посылки команд — объект CommandProcessor

CommandProcessor является основным объектом, обеспечивающим взаимодействие с пользователем. Его основная задача — преобразование событий в команды. При инициализации объекта задается набор команд и соответствующих им клавиш. Также указываются определяемые пользователем функции обработки фоновых задач, вызова справочной системы и специальные функции. После инициализации создается цикл обработки команд, который выглядит следующим образом:

```
Repeat
 Cmd := Commands.GetCommand(Key);
 Case Cmd of
  ccChar  :     {Введен символ}
  ccUp    :     {"стрелка вверх"}
  ccDown  :     {"стрелка вниз"}
...........
  ccUser0 :     {Пользовательская команда}
Until (Cmd=Quit)
```

Также после инициализации имеется возможность расширения набора команд с помощью метода AddCommand. Командный процессор позволяет обрабатывать события от клавиатуры и от мыши. Для вызова справочной системы необходимо обратиться к процедуре GetHelpProc и указать ее при вызове метода SetHelpProc. При вызове (через команду ccHelp) процедура GetHelpProc получает три аргумента, первый из которых определяет источник запроса, второй — объект-владелец командного процессора, а третий — контекст справочной системы.

Поскольку в Turbo Vision каждый отображаемый объект имеет свой собственный метод GetEvent, выполняющий преобразования событий в команды, реальным обработчиком этих событий является метод TApplication.GetEvent. Методы GetEvent остальных объектов просто вызывают GetEvent своих предшественников. В OPro каждый интерфейсный объект имеет свой командный процессор, который получает управление при активизации объекта.

Окна

Помимо средств для реализации окон и их компонентов: рамок, заголовков и полос прокрутки, OPro содержит ряд объектов для реализации "интерактивных" окон, которые созданы на основе абстрактного объекта CommandWindow. Основным методом этих объектов является метод Process, который выполняет обработку событий для данного объекта. Использование большинства отображаемых объектов (наследников объекта CommandWindow) можно свести к следующему коду:

```
TSomeObject.InitCustom(...)    {Инициализация}
TSomeObject.SetErrorProc(...)    {Обработчик ошибок}
TSomeObject.SetHelpProe(...)  {Справочная система}
.....
IsDone := False;
Repeat
 TSomeObject.Process;
 Case TSomeObject.GetLastCommand of
  ccXXX  :
  ccYYY  :
.........
  ccQuit :  IsDone := True;
Until IsDone;
TSomeObject.Done;              {Завершение}
```

Для каждого отображаемого объекта имеется возможность установки собственного обработчика ошибок и процедуры вызова справочной системы.

Меню

На мой взгляд, создание системы меню в OPro значительно проще и нагляднее, чем в Turbo Vision:

```
Var
 M : Menu;
...............
With M do
 Begin
  InitCustom   (....);          {Инициализация}
  mnOptionsOn  (....);          {Установка опций}
  mnOptionsOff (....);
  wFrame.SetFrameType(.....);    {Выбор типа рамки}
{Создание меню}
  AddItem(....);              {Горизонтальный элемент}
   AddFramedSubMenu(....);
   AddItem(....);              {Вертикальный элемент}
   AddItem(....);
   ItemsDone;
  AddItem(....);
   AddFramedSubMenu(....);
   AddItem(....);
   ItemsDone;
  ItemsDone;
 End;
```

Если и такой подход кажется сложным, можно использовать генератор меню — утилиту MAKEMENU, входящую в комплект поставки.

Имеется возможность использования различных типов рамок и изменения расположения меню. В отличие от Turbo Vision, меню в OPro может быть единственным элементом пользовательского интерфейса (то есть нет связки Меню — Рабочая область — Строка состояния).

Поля ввода

Основой для объектов, реализующих ввод данных, является абстрактный объект "поле" — наследник объекта DoubleListNode. В OPro поддерживаются следующие типы полей ввода:



Название	Тип информации
TextField	Текстовые поля без возможности редактирования
ArrayField	Поле для массивов символов
BCDField	Поле для BCD-типов
BooleanField	Поле для логических типов
ByteField	Поле для байтовых типов
CharField	Поле для символьных типов
ChoiceField	Поле для перечисляемых типов
CompField	Поле для комплексных типов
DateField	Поле для даты
DateStField	Поле для типа DateString
DoubleField	Поле для типа Double
ExtendedField	Поле для типа Extended
IntegerField	Поле для типа Integer
LongIntField	Поле для типа LongInt
MultiLineField	Альтернатива мемо-полям
NestedField	Поле для доступа к вложенным экранам
RealField	Поле для типа Real
ShortIntField	Поле для типа ShortInt
SingleField	Поле для типа Single
StringField	Поле для строчных типов
TimeField	Поле для времени
WordField	Поле для типа Word
YesNoField	Поле для логических типов (Yes/No)

Отметим, что поля ввода могут содержать шаблоны, которые хорошо знакомы пользователям dBase. В OPro также реализованы методы для преобразования значений полей и методы проверки введенной информации. Для облегчения создания программ, использующих поля ввода, поставляется утилита MAKESCRN, с помощью которой поля ввода располагаются на экране/экранах — создаются вводные формы, а затем эти формы могут быть сохранены в виде либо библиотеки (OPL), либо исходного текста для включения в программу.

Панель диалога

Объекты, реализованные в модулях OPDIALOG и OPDRAG, позволяют создавать панели диалога, соот-



ветствующие стандарту CUA/SAA фирмы IBM. Панель диалога может включать в себя кнопки, кнопки с зависимой и независимой фиксацией, кластеры, поля ввода и т.п. Специальный объект DragProcessor, наследник CommandProcessor, позволяет перемещать эти элементы и сами панели диалога по экрану. Для создания самой панели диалога используется объект DialogBox, после его инициализации элементы управления добавляются с помощью методов AddXXX, где XXX — название элемента управления:

Метод	Назначение
AddPushButton	Добавить кнопку
AddRadioButtons	Добавить группу кнопок
AddRadioButton	Добавить кнопку с независимой фиксацией
AddCheckBoxes	Добавить группу кнопок
AddCheckBox	Добавить кнопку с зависимой фиксацией
AddWindowControls	Добавить группу элементов управления
AddWindowControl	Добавить элемент управления
AddEditControl	Добавить строчный редактор

Каждый элемент управления посылает команду, которая обрабатывается методом Process панели диалога. Отметим, что внешним видом элементы управления практически не отличаются от элементов управления Turbo Vision.

Редакторы

Объект TextEditor представляет собой, как вы уже наверно догадались, текстовый редактор. Редактор в OPro позволяет выполнять блочные операции, операции поиска и замены, а также использовать "закладки" (bookmarker).

Для включения редактора в программу необходимо выполнить следующие действия:




```
Var
 TE        : TextEditor;
.........
TE.InitCustom(.....);
{Указать обработчики для строки состояния и ошибок}
TE.SetStatusProc(....);
TE.SetErrorProc (....);

 {Открыть редактируемый файл}
TE.ReadFile('FILENAME.TXT',FSize);

Repeat

  TE.Process;
{Обработка команд}
  ExitCommand := TE.GetLastCommand;

  Case ExitCommand of
   ccSaveExit,
   ccAbandonFile,
   ccError        :     AllDone := True;
  End;
Until AllDone;

TE.Erase; TE.Done;
......
```

Резидентные программы

Один из основных и комплексных модулей библиотеки — модуль OPro. В нем находятся подпрограммы управления обработкой прерываний (ISR). При помощи этого модуля возможно создание резидентных программ, которые могут автоматически определять наличие расширенной памяти (EMS) или виртуального диска и при необходимости выполнять свопирование. В приведенном ниже примере показано, насколько просто происходит создание резидентных программ с помощью OPro. Созданная резидентная программа автоматически свопирует себя в EMS или на диск. В памяти остается только ядро размером около 6 Кбайт, необходимое для загрузки оставшейся части резидентного приложения.

```
Program TSRDemo;
uses DOS,OPTSR,OPInt;
Const
 HotKey = $082C;          {ALT-Z}

{$F+}
Procedure PopUpEntryPoint(var Regs : Registers);
Begin
 Writeln('TSR Demo with Object Professional')
End;
{$F-}

Begin
 If DefinePop(HotKey, PopUpEntryPoint, Ptr(SSeg,SPtr),True) Then
 Begin
  Writeln('PopUp loaded, press <ALT><Z> to activate.');
  PopUpsOn;
  StayRes(ParagraphsToKeep,0);
 End;
 Writeln('Ошибка!');
End.
```

Работа со строками

OPro содержит все мыслимые и немыслимые функции для работы со строками: начиная от функций выравнивания и удаления лишних пробелов и заканчивая поиском строк в памяти. Поддерживаются функции

преобразования различных типов данных в шестнадцати-, восьмиричный или двоичный формат. Интересной является функция Soundex, позволяющая искать строки, похожие на заданную. Кроме того, имеется большое число функций для преобразования полных имен файлов. OPro содержит собственный модуль поддержки ASCIIZ-строк (строки, заканчивающиеся символом #0), которые по умолчанию могут содержать до 1024 символов. Практически все строчные функции могут быть использованы с ASCIIZ-строками.

Поддержка DOS-функций

Большинство функций, реализованных в модулс OPDOS, являются дополнениями к функциям стандартной библиотеки. Сюда включены функции, позволяющие управлять программой PRINT (спулером для вывода на устройства печати), осуществлять доступ к среде, функции чтения и записи секторов диска, функция вызова копии COMMAND.COM — ExecDos, которая предоставляет дочернему процессу максимальное количество памяти (за счет сжатия "кучи"), и многое другое.

Поддержка защищенного режима

По сообщению фирмы, переход к работе в защищенном режиме (PMode) не вызвал больших затруднений. Более комплексной была проблема изменения в системе управления кучей (heap) между версиями компилятора Turbo Pascal 5.5 и 6.0.

Для работы с функциями DPMI был введен новый модуль DPMI.PAS, в котором, помимо реализации вызовов практически всех функций интерфейса DPMI, определены несколько селекторов, которые могут быть использованы в приложениях. (Подробно о PMode-программах см. КомпьютерПресс № 2'93).

Кроме того, увеличение объема отладочной информации, необходимое для поддержки браузеров, привело к появлению нового модуля — OPCONST, в котором определены все константы, используемые модулем OPROOT.

Отметим, что модули поддержки резидентных программ не могут использоваться в PMode-программах, так как для работы последних требуется загрузка исполняющей системы (RTM.EXE) и сервера (DPMI16BI.OVL).

Документация

Отдельно — о документации. Разделенная на три тома, она содержит более 1600 страниц подробного описания всех объектов, процедур и функций, реализованных в библиотеке. Практически для каждой процедуры или функции приводятся примеры ее использования, а также ссылки на демонстрационные примеры.

Каждый тематический раздел (всего их 14) содержит введение, в котором не только описано назначение группы объектов/функций, но и приводятся теоретические сведения и разъясняется, почему был выбран именно такой способ реализации. Указываются также размеры внутренних структур, накладные расходы при использовании тех или иных свойств объектов, дается и другая полезная для разработчиков информация. Приложение содержит таблицу зависимости модулей, диаграмму иерархии объектов и полезные советы. Интересно отметить, что в приложении приводится список рекомендуемой литературы.

Object Professional или Turbo Vision?

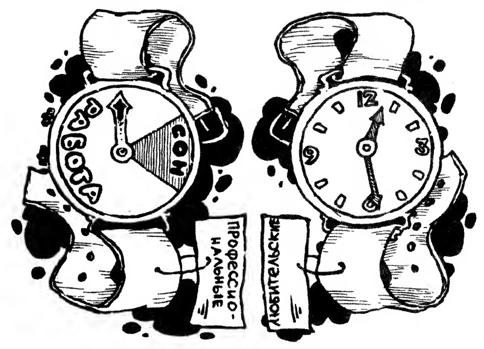
После появления библиотеки Turbo Vision (входящей в комплект поставки компиляторов Turbo Pascal версии 6.0 и 7.0) многие пользователи OPro в нерешительности встали перед затруднительным выбором: что лучше? Объективно оба средства являются объектно-ориентированными библиотеками для компилятора Turbo Pascal и предназначены для решения одних и тех же задач: создания пользовательских интерфейсов.

Отметим, что несмотря на всю кажущуюся общность, Turbo Vision и OPro построены на совершенно различных концепциях. В Turbo Vision реализована так называемая модель MVC — Monitor-View-Control, что архитектурно приближает эту библиотеку к среде Microsoft Windows.

OPro содержит большой и взаимосвязанный набор объектов высокого уровня, готовых к использованию. В отличие от Turbo Vision, в OPro не используется механизм управления событиями, что, с одной стороны, делает использование этой библиотеки более простым, но с другой — затрудняет перенос приложений в среду Microsoft Windows (идеологически Turbo Vision соответствует Windows-библиотеке ObjectWindows фирмы Borland). Также Turbo Vision не содержит объектов для взаимодействия с операционной системой, работы с расширенной и дополнительной памятью, поддержки вывода на устройство печати и т.п. Создание резидентных программ с помощью Turbo Vision также затруднено по ряду объективных причин.

Отказ фирмы от поддержки механизма обработки событий (в том виде, в котором он присутствует в Turbo Vision) объясняется тем, что OPro — это объектно-ориентированный вариант библиотеки Turbo Professional. Механизм обработки событий не является необходимым для DOS-программ, и его использование может вызвать затруднения у неподготовленных программистов. В большинстве случаев использование такого механизма приводит к замедлению работы программы, что вызвано дополнительной диспетчеризацией событий.

Turbo Vision позволяет довольно легко создавать приложения, по внешнему виду и интерфейсу напоминающие продукты фирмы Borland, но эта библиотека не достаточно гибка, если требуются какие-либо не-



стандартные элементы. Подход "оболочки приложения" (application framework) требует от программиста больших начальных знаний о системе, для того чтобы начать создавать свои приложения, и накладывает определенные ограничения. Помимо этого, библиотека OPro включает в себя ряд дополнительных возможностей: редактор экранов, возможность распечатки форм, наличие средств просмотра баз данных и т.п.

Отметим, что большинство модулей, не относящихся к интерфейсным модулям, могут быть с успехом использованы в Turbo Vision-программах. В первую очередь, это модуль OPDOS, а также ряд других модулей (см. табл.).

В завершение хочется отметить, что даже если вы не используете всех возможностей OPro, ознакомление с документацией и исходным текстом модулей может быть очень полезно для расширения кругозора и изучения различных аспектов использования компилятора Turbo Pascal.

Несколько цифр, позволяющих оценить объем этой библиотеки: более 50 модулей, которые включают в себя порядка 130 объектов, свыше 2000 документированных методов, процедур и функций, а также более 1000 дополнительных процедур, используемых внутри этих объектов. Документация, объем которой превышает 1600 страниц. Библиотека OPro вобрала в себя более чем 8-летний опыт работы по созданию библиотек для компилятора Turbo Pascal. С момента создания было продано более 25 000 копий библиотек Turbo Professional и Object Professional, а также несколько тысяч копий библиотеки OPro для C++.

Таблица. Модули, которые можно использовать совместно с Turbo Vision

Модуль	Назначение
OPBCD	Поддержка BCD-арифметики
OPCOLOR	Символические определения цветов
OPDATE	Функции представления даты/времени в различных форматах
OPDEVICE	Функции вывода на устройства печати
OPDOS	Расширенные функции управления DOS
OPEMS	Функции управления расширенной памятью
OPEXTMEM	Функции управления дополнительной памятью
OPFORM	Создание форм
OPINLINE	Различные системные макросы
OPINT	Управление прерываниями
OPKEY	Символические определения клавиш
OPLARRAY	Функции управления массивами большого объема
OPPRINT	Функции вывода на устройства печати
OPPRNLOW	Функции вывода на устройства печати
OPROOT	Объекты-контейнеры
OPSORT	Функции сортировки
OPSTRING	Функции управления строками
OPSWAP	\
OPSWAP1	— Создание TSR-программ
OPTSR	/
OPXMS	Функции управления дополнительной памятью

Требования к системе

Компьютер IBM PC, XT, AT, компилятор Turbo Pascal версии 5.5 и выше, 640 Кбайт оперативной памяти, жесткий диск (при полной установке библиотека занимает более 11 Мбайт), для изменения ряда модулей требуется TASM версии 1.0 и выше или MASM версии 4.0 и выше.

Поставка

Настоящая версия OPro (1.20) поставляется на трех дискетах 5,25 дюймов объемом 1,2 Мбайт и на пяти дискетах 3,5 дюймов объемом 720 Кбайт. Документация располагается в трех томах. Дополнительно на дискетах поставляется описание отдельных модулей. Файл README.1ST содержит описание изменений, внесенных в текущую версию библиотеки.

Официальным представителем Turbo Power Software в России является фирма "Лэинтек" (С.-Петербург), которая осуществляет поставки через дилерскую сеть.

А.Федоров



—Я не старался научить читателя, как пользоваться чужими подпрограммами, мне хотелось научить читателя, как составлять лучшие подпрограммы самому.

Дональд Кнут
"Искусство программирования для ЭВМ"

Назад в будущее,

или Кое-что о классике предмета

Библия для программистов

Более двадцати лет назад появился первый том монографии Дональда Кнута "Искусство программирования для ЭВМ" (Donald E. Knuth, "The art of computer programming"). Сегодня вышло уже шесть томов, и автор продолжает работу (в русском переводе были изданы первые три тома [1]). Задачи беспрецедентной по объему материала и глубине изложения монографии автор определил ясно: служить справочником для специалистов и учебником для всех желающих выработать у себя умение рационально мыслить о программах.

Если как справочник по алгоритмам книга Дональда Кнута обладает одним из самых высоких индексов цитирования в мировой литературе по программированию, то как учебник она используется редко и не по прямому назначению, несмотря на то что именно "учебная" часть занимает по объему гораздо большее место, чем собственно готовые алгоритмы. Автор "Искусства программирования..." подчеркивает непохожесть своего учебника на другие.

Почему уникальный учебник по программированию не стал массовым общедоступным инструментом для воспитания мышления программистов? Может быть, он устарел? В чем отличие книги Дональда Кнута от большинства учебников по программированию? Как вообще возможно развить навык рационального мышления?

Дональд Кнут — "чистый" математик, пришедший в программирование. Как математик, он хорошо знаком с технологией выработки навыков рационального мышления у математиков. Несколько поколений специалистов "королевы наук" выросло на книгах выдающегося математика и педагога Дьердя Пойа [2]. Сущность предлагаемой Пойа техники состоит в правильном подборе серии задач, которые нужно решать последовательно. Если пытаться решить сразу самую сложную задачу серии, шансов на успех практически нет даже у очень способного новичка. Сначала решаются несколько простых задач, вводящих в элементарные понятия о предмете. Затем задачи немного усложняются, но имеющийся опыт позволяет решить их достаточно быстро. Освоившись на этом уровне сложности, можно перейти на следующую ступеньку, где человеческий интеллект должен опять "потоптаться", прежде чем шагнуть дальше. В конце серии способный ученик решает задачи, являющиеся современными исследовательскими проблемами, подступиться к которым без постепенной выработки соответствующих интеллектуальных навыков он не смел бы и мечтать. Для каждой темы, для каждого вопроса требуется своя серия "подводящих" к решению задач — и огромная соответствующая работа по подготовке задачника. Свобода мышления учащегося, его индивидуальность при таком методе развиваются в максимальной степени вместе с навыком рационального мышления, который есть не что иное, как опыт, результат деятельности по решению задач. Разница между случайными задачами и за-

дачами, подобранными в специальные последовательности, состоит в скорости, устойчивости и глубине формирования навыка рационального мышления.

Для "Искусства программирования..." принципы Пойа являются фундаментальными. Вспомним и содержание упражнений, занимающих бóльшую часть толстых томов серии, и градации сложности упражнений, введенных автором, и то, что среди упражнений периодически встречаются нерешенные исследовательские проблемы. Это совсем не случайно! Только на первый взгляд книги Кнута можно принять за справочник по алгоритмам. Готовые алгоритмы — надводная часть айсберга. Основное содержание, смысл и цель этих книг — реализация программы по воспитанию дисциплины ума программиста на специально (и прекрасно) подобранных сериях практических задач! В течение долгих лет Дональд Кнут выполнял для нас работу, объем и значение которой трудно переоценить, в то время как его современники доказывали пагубность оператора GO TO, придавали выписыванию текста программы "лесенкой" или ограничению числа используемых конструкций характер религиозных догматов. Дональд Кнут сделал для программирования очевидные с точки зрения математика вещи, полагая, что, как и в любой другой области человеческой деятельности, дело здесь в развитии человеческих способностей, которые только и дают возможность эффективно использовать самые изощренные формальные технологии программирования [3, 4].

О качествах работы Дональда Кнута мы можем судить по еще одному характерному примеру. Двадцать лет назад можно было только гадать о характере развития аппаратуры и программного обеспечения. Сегодня, рассмотрев архитектуру машины MIX, введенной в качестве основной машины серии книг Кнута, можно лишь удивляться тому, как точно угадал автор аппаратное ядро практически любой современной ЭВМ. Эволюция машины фон Неймана происходит почти по тем же законам, по которым развивается математическая теория из однажды выбранной системы аксиом. Даже модификация аксиоматики приводит не к исчезновению основных черт классической машины, а к изменению пропорций и роли ее частей.

То же касается ассемблера MIXAL, все приемы программирования на котором прекрасно вписываются в современные системы программирования на машинных языках, вплоть до идеи переопределяемых временных меток. Осваивая MIX/MIXAL, мы знакомимся не просто с "учебной" системой, а с характерным набором аппаратно-программных черт современных массовых компьютеров. Этот пример наглядно показывает эффективность применения строгого мышления там, где принято делать туманные намеки и злоупотреблять словами "возможно" и "вероятно". Работа Дональда Кнута построена на строгой логике даже там, где ее автор мог бы опираться на личные вкусы и пристрастия. Поэтому можно утверждать, что фундаментальный труд Дональда Кнута будет активно использоваться во все времена, пока будут существовать программирование и программисты. Говорить, что книга Дональда Кнута устарела, можно только не понимая, что она не может устареть, как не могут устареть "Начала" Евклида.

Книга Дональда Кнута как классический комплекс средств для развития мышления программистов не просто одна из лучших, — она единственная. Ее корни глубоко уходят в математическую культуру, при этом сохраняется доступность для новичка, не знающего математики. Можно даже утверждать, что древний храм математики приобрел еще одни ворота, через которые в "чистую" математику могут и должны входить программисты, не имеющие специальной математической подготовки. Для начала от них не потребуется ничего, кроме готовности учиться писать хорошие программы.

Технологическая петля

И все-таки использование "Искусства программирования для ЭВМ" как учебника, а не как справочника, встречает целый ряд препятствий. Попытки создать условия для работы с книгой как с учебником неоднократно предпринимались ранее, но к успеху не привели. Прежде чем делать еще одну попытку, следует разобраться в характере и причинах затруднений.

Проблема первая, техническая

Использование книги Дональда Кнута как учебника невозможно без реализации специальных программ поддержки. Как минимум, это эмулятор машины MIX и ассемблер MIXAL. Такая работа выполнялась уже неоднократно, в разное время и на разных машинах, почти с момента появления первого тома 20 лет назад. Технический уровень исполнения работ не позволил создать массовую, доступную каждому желающему современную систему, оставляя потенциал книги Кнута скорее теоретическим соблазном, чем практическим средством. Положение изменилось сравнительно недавно, с появлением интегральных сред поддержки разработки программ, и инструментов, помогающих строить такие среды.

Интегральная среда, доступная любому пользователю, содержащая в себе полный набор средств для работы с книгой Дональда Кнута как с учебником, является программной системой высокой сложности, насчитывающей 40-50 тыс. строк исходного текста на языках С или Pascal. Создавать такую среду "с нуля" — задача очень сложная. К счастью, фирма Borland, выпустив на рынок свои продукты Borland C/C++ и Turbo Pascal 6.0, придала к ним средство Turbo Vision — конструктор интегральных сред с возможностями расширения средствами, потребными пользователю. Использование Turbo Vision в качестве ядра учебной системы резко снижает трудоемкость разработки уникальных средств, необходимых для поддержки книги Кнута как учебника. Но даже такая мощная поддержка не делает решение задачи тривиальным.

Проблема вторая, технологическая

Современное программирование имеет сильный методический крен в сторону чисто технологических аспектов. Исторически сложилось так, что в программировании был создан культ инструментов и технических приемов. Личность программиста, оригинальность мышления, стиль, вкус, индивидуальные черты — все было объявлено второстепенным, а чаще — просто вредным, ради великой цели: внедрения индустриально-поточных методов в программирование. Мы перешли на языки высокого уровня, стали программировать модульно, структурно, теперь — объектно. Нам каждый раз обещали качественный рост производительности труда, эффективное устранение ошибок, читабельные тексты программ и прочие приятности. Сбылись ли обещания? Скорее нет, чем да.

Программирование стало массовой профессией, но написание каждой серьезной программы так и остается задачей индивидуальной либо идет под жестким контролем одного разработчика, стирающего другие индивидуальности. Никакого конвейера не получилось: возможности создания программных полуфабрикатов, стандартизации интеллектуальных продуктов оказались не такими широкими, как хотелось бы. Даже при объектном подходе программа все равно пишется каждый раз заново — с начала и до конца. Использование ранее разработанных объектов удобнее, чем ранее разработанных подпрограмм, но вовсе не в такой решающей степени, как обещает реклама. Мы получили прекрасные инструменты, но эффективно применять эти инструменты могут немногие, тогда как большинство продолжает безнадежно вращаться в унылом цикле "редактор-компилятор-отладчик".

Мы переоценили технологическую часть программирования. Точнее, мы приняли технологию за собственно программирование. И сейчас как абстрактная философия выглядит предложение обратиться, наконец, к личности программиста, развитию приемов его мышления, личных навыков, качеств, особенностей, умения рационально мыслить о программах.

Проблема третья, психологическая

Нам труднее всего решиться изменить себя. Мы избегаем этого до последнего, предпочитая менять версию компилятора. Даже если речь идет об осознанном улучшении личных способностей, мы бы предпочли вместо этого использовать какую-нибудь подходящую программу. Человеческий консерватизм, особенно консерватизм опытных специалистов, был бы оправдан и в нашем случае, если бы не приводил к жесткому самоограничению профессиональных возможностей. Тем более, что мы все равно меняемся — с каждым новым инструментом, технологией, изученным текстом чужой программы. Но меняемся случайным, не зависящим от ясной цели и нашей воли образом. И в результате такого "случайного блуждания" мы получаем вовсе не то, что обещает реклама очередного "патентованного средства".

Лень часто стремится выглядеть респектабельно, рядиться в одежды "большого практического опыта". Учебник Дональда Кнута предлагает программу интенсивной интеллектуальной работы по развитию способностей любого программиста — и новичка, и профессионала. Первоначальная психологическая реакция программиста на предложение заняться собственными способностями вместо ожидания технологического чуда будет, вероятно, отрицательной.

Выводы

Первая (техническая) проблема не объясняет неприятия книги Дональда Кнута как учебника. Ее решение хотя и не просто, но вполне реально. Вторая и третья проблемы создают барьер гораздо более серьезный, который можно назвать "технологической петлей" программирования. Психологическая и технологическая традиции пытаются свести программирование к чисто техническим его аспектам, исключая из рассмотрения человеческую сторону проблемы. Эволюция программирования объявляется эволюцией технологии программирования. Личные творческие способности программиста игнорируются или с ними даже борются, объявляя их вредными. Перспективой программирования считается появление чудесной технологии, полностью автоматизирующей процесс создания программы, делающей программиста вообще ненужным, удаляя его из программирования навсегда.

Никто не станет отрицать важности и необходимости технологии, полезности технических приемов. ***Но ПРОГРАММИРОВАНИЕ — НЕ ТЕХНОЛОГИЯ, а АВТОРСКОЕ ИСКУССТВО. Технология занимает почетное, но не первое место. Решающий фактор программирования — личные творческие способности программистов.***

Игнорируя человеческую сторону программирования, мы внушаем способным новичкам иллюзию, что программистом можно стать, научившись писать программу из нескольких сотен строк на языке Pascal, пользуясь каким-либо технологическим приемом, вроде структурного кодирования, и профессиональный рост далее сводится к увеличению размеров создаваемых программ. Этот опасный (само)обман приводит на практике к жестоким разочарованиям и огромным невозвратимым потерям времени в наиболее эффективном для обучения возрасте. Выйти из технологической петли мы можем только осознавая тупиковый, иллюзорный характер ее целей.

Чтобы разорвать порочный цикл "косвенных" технологий, нам нужно много программистов, хороших и разных. Стать *СОБОЙ* может каждый программист, но для этого он должен всерьез развивать *КАЧЕСТВЕННУЮ* сторону своих способностей. Любой, кто решается это делать, понимает, что серьезной альтернативы интеллектуальному тренажеру Дональда Кнута сегодня

просто нет. Конечно, это программа работы почти на всю профессиональную жизнь. Но легкие пути к совершенству обещает только плохая реклама.

Как видно, игнорирование самой существенной части работы Дональда Кнута далеко не случайно. Жить ли с иллюзией прогресса или действительно развиваться — личное дело каждого. Однако выбор должен быть доступен всем.

Краткий обзор Turbo MIX

Чтобы обеспечить (сейчас и каждому!) возможность работать с книгой Дональда Кнута как с учебником, российской фирмой VICTORIA International была создана интегральная среда Turbo MIX.

Это оболочка, почти полностью подобная оболочкам Borland, сопровождающим такие языки, как C/C++ и Pascal. Вместо этих языков оболочка Turbo MIX содержит эмулятор машины MIX и ассемблер MIXAL. Оболочка включает в себя стандартный многооконный Turbo-редактор (с объемом редактируемого файла до 64 Кбайт), слегка измененную систему меню и диалоговых окон, встроенную систему Help, встроенный символьный отладчик, подобный отладчику Turbo Debugger, загрузчик программ вместо компоновщика, набор стандартных приложений (калькулятор, календарь, таблицы кодов MIX и ASCII). Ядро системы составляют эмулятор расширенной машины MIX и расширенный ассемблер MIXAL. Кратко рассмотрим все компоненты.

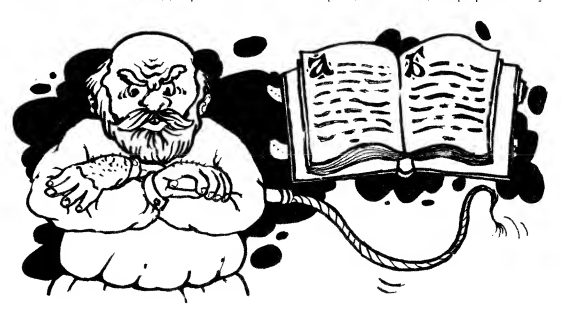
Текстовый редактор является стандартным редактором, поставляемым в конструкторе Turbo Vision. Он очень похож на редакторы, применяемые в собственных системах Borland, с небольшими отличиями. Самое существенное отличие состоит в использовании файла объемом не более 64 Кбайт. Для работы с книгой Дональда Кнута это ограничение несущественно. Но даже если вам захочется написать очень большую программу, вы сможете это сделать, поскольку ассемблер MIXAL снабжен соответствующими средствами. Кроме того, стандартный редактор Turbo Vision был несколько модифицирован функционально (например, расширением функций поиска).

Система меню и диалоговых окон сделана максимально похожей на применяемую в оригинальных инструментах Borland. Сохранен смысл всех названий, горячих клавиш и кнопок, за исключением тех, что связаны с введением специфических особенностей MIX/MIXAL.

Встроенный Help работает так, как это предусмотрено в Turbo Vision — несколько более грубо, чем в оригинальных продуктах Borland. Однако смысловая часть, контекстные подсказки, индекс и прочие справочные элементы присутствуют в полном объеме.

Встроенный отладчик очень похож на Turbo Debugger. Отличия есть в механизме обслуживания контрольных точек (в первой версии нет условных контрольных точек и счетчиков числа проходов по точкам), watch-переменные имеют типы, принятые в MIX/MIXAL, и, самое существенное, отладчик может работать не только с исходным текстом программы, но и с ассемблерным листингом. Работающие на ассемблерах знают, что листинг при отладке гораздо информативнее исходного текста, так как содержит кодовую колонку и различные пометки трансляции. Что касается режима работы, то здесь встроенный отладчик Turbo MIX работает более в манере автономного отладчика Turbo Debugger, чем встроенного интегрированного отладчика C/C++ или Pascal (например, контрольную точку можно установить, только когда программа уже загружена в память машины MIX).

Поскольку ассемблер MIXAL готовит сразу исполняемый файл, компоновщик программ не нужен. Вместо



него Turbo MIX содержит загрузчик, который читает готовую программу с диска и помещает ее в память машины MIX. Допустимы два режима загрузки: основной, когда память машины MIX обнуляется перед записью программы, и дополнительный, когда при загрузке программы содержимое памяти не изменяется, за исключением слов, непосредственно принадлежащих загружаемой программе. Во втором случае обеспечивается режим ручной "склейки" в памяти машины MIX произвольного кода.

Стандартные приложения включают в себя подсистемы, придаваемые программе в Turbo Vision — календарь, простейший десятичный калькулятор, таблицу кода ASCII. Дополнительно приложения содержат таблицу внутреннего алфавитно-цифрового кода для машины MIX, введенного Дональдом Кнутом, и подсистему вызова других программ, необходимых пользователю, как приложений Turbo MIX.

Эмулятор машины MIX является аппаратным ядром системы. Через окна процессора, памяти и устройств аппаратура машины MIX может быть доступна пользователю непосредственно. Однако машиной MIX можно пользоваться и "вслепую", как обычной машиной, прогоняя на ней с помощью стандартной команды Run свои программы.

Turbo MIX реализует расширение машины MIX, описанной Дональдом Кнутом. Расширение содержит все средства стандартной машины MIX, добавляя некоторые полезные черты. Например, память машины MIX снабжена тэгами. Тэг слова памяти отмечает тип данных, хранимых в слове: команда, целое число, текст, неопределенное значение. Процессор расширен регистром команд, с которым связаны регистр исполнительного адреса и регистр текущего операнда по исполнительному адресу. Это позволяет наглядно наблюдать покомандное исполнение программы процессором даже без привлечения отладчика. Машина снабжена полным набором периферийных устройств: ленты, диски, ввод/вывод перфокарт, принтер, ввод/вывод перфоленты. Все устройства моделируются обычными файлами в файловой системе IBM PC. Отсутствующие диски создаются во время работы, по требованию. Эмулятор имитирует временну́ю диаграмму работы устройств, соблюдая задержки операций ввода/вывода. Размеры дисков, длины блоков, задержки, имена файлов-устройств и прочие параметры эмулятора пользователь может задавать сам через стандартный механизм Options.

Основным языком программирования в Turbo MIX является ассемблер MIXAL. Аппаратные расширения машины MIX получили соответствующее отражение и в ассемблере. В частности, листинг содержит колонку тэгов. Ассемблер MIXAL создает исполняемую программу, которая пишется либо непосредственно в память машины MIX, либо в файл на диске для последующей загрузки и использования (флаг Destination). Ассемблер вырабатывает листинг, который может снабжаться таблицей имен и списком перекрестных ссылок. Листинг, наряду с исходным текстом, может быть использован отладчиком. Для написания программ, состоящих из более чем одного файла, в MIXAL включена препроцессорная команда вставки файла в поток трансляции.

Еще одна принципиальная особенность Turbo MIX — отсутствие в его работе двоичных файлов. Все файлы, создаваемые или используемые системой, — текстовые. Исполняемый файл, вырабатываемый ассемблером, файлы всех устройств (текстовых и не текстовых), таблицы отладчика — это обычные ASCII-файлы, принятые на IBM PC. При вводе текстовой информации через устройства ввода она переводится из кода ASCII во внутренний MIX-код. При выводе файлы формируются в коде ASCII. Это существенно замедляет работу ассемблера и эмулятора, но обеспечивает пользователю абсолютный контроль над ситуацией без необходимости разбирать нечитабельный двоичный код. Чтобы посмотреть или отредактировать любой файл, достаточно использовать обычный текстовый редактор или программу просмотра текста, если файл слишком большой по размерам (например, диск), через механизм выхода в подсистемы или MS-DOS.

При создании Turbo MIX предпринимались сознательные усилия по обеспечению максимального соответствия работе в инструментальных системах Borland C/C++ или Turbo Pascal 6.0. Делалось это для того, чтобы пользователь, имеющий опыт работы с системами Borland (а таких — подавляющее большинство), легко освоил Turbo MIX без каких-либо "переучиваний". Это сильно снижает психологический барьер при освоении новой системы. Новичок, у которого Turbo MIX может стать первой серьезной системой, получает после работы на нем бесплатный подарок — навык работы в оболочке, общей для всех инструментов Borland (в том числе применяемых при программировании на языках C/C++ и Pascal).

Turbo MIX призван полностью снять все технические проблемы, возникающие при использовании книги Дональда Кнута как учебника. Наличие Turbo MIX переводит проблему в плоскость свободного выбора каждого



осведомленного программиста или новичка, желающего приобщиться к нашей профессии.

Инструменты развития способностей

Книга Дональда Кнута с приложением типа Turbo MIX является типичным примером интеллектуального продукта, ориентированного на развитие человеческих способностей.

Принято считать, что играть любят только дети. Но компьютерные игры потеснили футбол и хоккей даже в сознании взрослых. С компьютером играют все. Интерес вызывает игра, в которой присутствует какой-то чисто человеческий стимул: честолюбие, азарт, стремление к успеху, состязательность. Однако программисты в массе своей не азартны и не склонны к постановке рекордов. Немногие программисты любят регулярно сидеть перед компьютером, играя в F-19 или Prince of Persia. Нет времени, много работы, а игра означает впустую растраченное время.

Вот если бы существовали игры, которые помогали бы лучше делать повседневные дела, "играючи" решить сложную проблему, которые делали бы человека умнее, способнее с каждым сеансом! И большинство захочет, чтобы это были не шахматы, потому что нужно стать не вообще умнее где-нибудь, а умнее в конкретной области.

Программистам повезло, потому что такая игра для них есть, и появилась она за несколько тысяч лет до появления компьютеров. Это математика. Великая интеллектуальная дисциплина с жесткими правилами и критериями истины, но одновременно живое искусство, окрашенное чувствами, эмоциями, азартом поиска истины. Творцы математики, относясь к ней вполне серьезно, не раз указывали на ее "игровые" качества.

Хотим мы этого или нет, осознаем или предпочитаем не видеть, но программирование — "числовое" и "нечисловое" — стоит на прочном фундаменте математических идей, понятий, принципов. Программирование связано с математикой гораздо теснее, чем любая дисциплина, даже теоретическая физика. Казалось бы, программисты должны иметь серьезную математическую подготовку, не уступающую по крайней мере в навыках строгого мышления профессиональным математикам. Увы, это совсем не так. В массе программисты не имеют серьезной математической подготовки и возможности (желания) ее получить. Это приводит к популярности и массовому распространению не самых лучших, а самых простых и доступных (до примитива) технологий программирования. Только один пример. Мы мучаемся с ошибками, тогда как уже около двадцати лет назад создана технология написания программ без ошибок [4]. Неужели ее применение на практике требует создания особо сложных инструментов, невозможных сегодня? Ничего подобного. Она требует других программистов — с развитыми навыками формального мышления, которые они готовы применять каждый день, на работе.

Разумеется, невозможно всем нам сейчас же забросить свои повседневные дела и сесть за парты — учить математику, хотя тем, кто учится сейчас, имеет смысл крайне серьезно отнестись к этому предмету. Но все мы, даже самые занятые, имеем реальные паузы в работе, да и просто возможность каждый день выделять хотя бы час для того, чтобы запустить не ТЕТРИС, а что-то вроде Turbo MIX, открыть книгу, и все-таки попытаться разобраться в том, как математики преодолевают уже много сотен лет наши проблемы.



Инструменты развития способностей должны помогать пользователям достигать максимума своего профессионального потенциала, тренируя или развивая специфические профессиональные качества, приемы мышления, интеллектуальные стереотипы. Это формирующееся направление выглядит сегодня неоднородно. К нему можно отнести некоторые экспертные системы, развивающие игры, компьютерные тренажеры. С этой точки зрения Turbo MIX — не единственная в своем роде программа. Хотелось бы верить, что нам будут доступны многие программы, образующие класс инструментов

развития способностей, — свои для бухгалтеров, юристов, химиков или архитекторов. Программисты благодаря великой книге Дональда Кнута имеют временное преимущество.

Вместо заключения

Есть три способа разрешения проблем. Способ первый состоит в том, чтобы попытаться радикально изменить мир. Мы хорошо знаем, куда ведет этот путь.

Второй способ состоит в том, чтобы ничего не делать и ждать, пока все само собой образуется. В нашем случае нет оснований считать, что программирование преодолеет технологическую петлю путем постепенного прогресса — мы можем до этого не дожить. Основанием такому утверждению служит проводимая сейчас подготовка более обширного вторжения в программирование объектных и непроцедурных методов, развитие нейрокомпьютерных систем. Нас готовят к очередному витку технологической петли, который аккуратно перенесет все наши сегодняшние проблемы на новый уровень сложности. Новые технологии — это прекрасно! Если они действительно разрешают уже имеющиеся проблемы. "Новыми" они могут называться и потому, что добавляют к существующим проблемам более сложные.

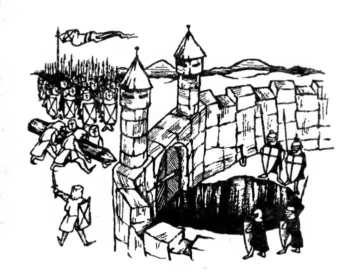
Третий способ заключается в том, чтобы начинать изменения с себя и постепенно (с помощью не используемых сейчас из-за отсутствия достаточной квалификации эффективных формализмов) обрести способность решать любые существующие задачи.

Во всяком случае, у нас есть выбор.

А.Седов

Литература:

1. Дональд Кнут. Искусство программирования для ЭВМ.
   Том 1. Основные алгоритмы. — М.: Мир, 1976.
   Том 2. Получисленные алгоритмы. — М.: Мир, 1977.
   Том 3. Сортировка и поиск. М.: Мир, 1978.
2. Дьердь Пойа. Как решить задачу. — М.: 1961.
   Дьердь Пойа. Математическое открытие. — М.: Наука, 1970.
   Дьердь Пойа. Математика и правдоподобные рассуждения. — М.: Наука, 1970.
   Г.Полиа, Г.Сеге. Задачи и теоремы из анализа. В 2-х т. — М.: Наука, 1978.
3. Knuth D.E. Structured Programming with GO TO Statements. Computer Serveys, 6(4), pp.261-301 (1974).
4. Э.Дейкстра. Дисциплина программирования. — М.: Мир, 1978.
   Д.Грис. Наука программирования. — М.: Мир, 1984.



Организация противодействия анализу принципов работы и логики защитных механизмов — одна из важнейших задач, стоящих перед разработчиками систем защиты от несанкционированного копирования.

Регистрация работы отладчика

Согласно принятой классификации [1,3], исследование программ может быть статическим — с помощью дизассемблеров и динамическим — с помощью отладчиков и эмуляторов. Защита от дизассемблирования, как правило, достаточно успешно выполняется с помощью шифрования кода, что в общем случае не гарантирует защиту от динамического исследования. Существуют собственные методы противодействия динамическому анализу программ, при реализации которых основной проблемой является определение присутствия отладчика. В настоящее время известны следующие способы решения данной задачи [1-4]:

- регистрация искажения кода программы в присутствии отладчика при нестандартном расположении стека либо при использовании конвейерного принципа обработки команд центральным процессором;
- подсчет контрольной суммы участка кода на предмет выявления точек останова;
- проверка действительного запрещения аппаратных прерываний;
- измерение времени выполнения участков программы;
- обнаружение взведенного Т-флага через потерю трассировочного прерывания.

Как видно из приведенного краткого обзора методов обнаружения отладчиков, подавляющее их большинство срабатывает только при пошаговом выполнении программы, тогда как практически не описаны методы обнаружения, независимые от режима работы отладчика. Рассмотрению таких методов и посвящена данная статья.

Несмотря на объявленную тему, было очень трудно удержаться от соблазна привести один любопытный способ обнаружения отладчика, основанный на пошаговом выполнении программы. В PSP программы со смещением 2Eh находится 4-байтная область, в которую DOS помещает указатель стека при обращении к системным функциям (во время выполнения ряда системных вызовов — определить их перечень можно только экспериментально — используется внутренний стек DOS). При пошаговом выполнении первоначальную обработку 21h прерывания производит отладчик, чей указатель и будет сохранен в активном PSP. Обнаружение отладчика, основанное на сравнении указателей стека, иллюстрирует несложная программа, которая в случае пошагового выполнения выдает сообщение “Hello, debbuger!“.

```
        .MODEL  SMALL
        .STACK  100h
        .DATA
Greeting    db  'Hello$'
Debug       db  ', debugger!',0Ah,0Dh,'$'
        .CODE
Start:
        mov     ax,@data
        mov     ds,ax
        mov     ah,09h
        mov     dx,OFFSET Greeting
        int     21h
        mov     ax,ss
        cmp     ax,es:[30h]
        je      Exit
        mov     ah,09h
        mov     dx,OFFSET Debug
        int     21h
Exit:
        mov     ax,4C00h
        int     21h
        END     Start
```

Прежде чем перейти к изложению основного материала, хотелось бы сделать ряд замечаний. В дальнейшем под операционной системой будет пониматься MS-DOS версии 5.0, а под отладчиками — Turbo Debbuger (как TD 2.5, так и TD286 2.5; они имеют



различия с точки зрения данной статьи), CodeView 3.0 и штатный системный отладчик Debug. Вполне вероятной может оказаться ситуация, когда для других версий этих продуктов признаки присутствия окажутся недействительными. Но проверка всех возможных сочетаний версий потребовала бы слишком большого времени. И еще. Данная статья ориентирована в первую очередь на профессионалов, имеющих доступ к необходимой справочной информации, поэтому здесь не приводится полное описание используемых системных структур.

Итак, начнем с самого начала, а именно с начального состояния регистров. Как было указано в [5], оно должно быть следующим: AX=BX, что отражает корректность имени диска аргументов командной строки, DX=DS, SI=IP, DI=SP. К сожалению, авторы [5] допустили пару неточностей. Во-первых, AX=BX отражает корректность имени не для одного, а для двух аргументов командной строки [6], причем AL индицирует корректность первого аргумента, а AH — второго. Соответственно AX, равно как и BX, может принимать четыре значения: 0000, 00FF, FF00 и FFFF. Здесь важно отметить тот факт, что отладчики не поддерживают BX=AX. Поэтому, если AX не равно нулю и BX не равно AX, можно делать вывод о присутствии отладчика.

Во-вторых, что является более существенным признаком, значение регистра CX определено и равно длине образа программы, загружаемого в память. Если длина образа превышает 64 К (то есть больше, чем можно описать одним 16-разрядным числом), то CX содержит остаток от деления длины на 2^{16}. Для COM-файла значение CX равно его длине, для EXE-файла оно вычисляется на основании данных, полученных из заголовка, по следующей формуле:

```
CX = { (PgCnt-1)*200h-HdrSize*10h+LastPg     при LastPg>0
     { PgCnt*200h-HdrSize*10h                при LastPg=0,
```

где PgCnt — длина образа в 512 (200h)-байтных страницах, включая заголовок; HdrSize — размер заголовка в 16-байтных параграфах; LastPg — длина последней неполной страницы.

Несмотря на то что практически все регистры общего назначения — CX, DX, DI и SI (об AX и BX было сказано выше) — при входе в программу имеют содержательное значение, отладчики, по какой-то тайной причине, предпочитают их обнулять. Поэтому для тех систем защит, которые реализованы в виде пристыковочных модулей и получают управление перед защищаемой программой, есть смысл проверять начальное состояние регистров, причем делать это, как вы сами понимаете, можно самым изощренным способом.

Кроме регистров общего назначения, можно проверять значение регистра флагов, которое должно быть равным 0202h, поскольку, например, TD286 загружает программу со значением 3202h.

Вообще создается впечатление, что у каждого отладчика свои "отпечатки пальцев". У Debug они обнаруживаются при формировании пути загрузки в окружении DOS для исследуемой программы. Как знает любой начинающий пользователь, полный путь должен состоять из логического имени устройства, пути и имени файла с расширением. Даже если программа была запущена из корневой директории, путь загрузки содержит по меньшей мере имя логического диска и имя файла. Debug же помещает в окружение DOS только имя файла. Простите, но поверить в то, что работающая программа не имеет пути загрузки, это все равно, что поверить в непорочное зачатие.

А вот как "следит" CodeView. В PSP программы начиная со смещения 18h располагается область длиной

20 байт, в которой хранятся номера handle'ов (по байту на handle), доступных данной программе. При автономном запуске программе доступны пять устройств, handle'ы которых помещены в PSP. Оставшиеся 15 handle'ов зарезервированы значением 0FFh. Несмотря на этот стандарт, CodeView запускает программу с 6 открытыми и соответственно 14 зарезервированными handle'ами.

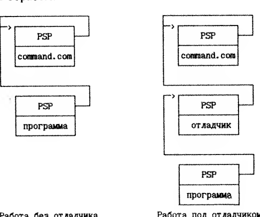
Как вы могли отметить, все описанные признаки работы отладчиков размещаются в программе и принадлежащих ей системных областях данных (PSP и окружение DOS). Другое место, где можно поискать признаки работы отладчиков, — это операционная среда программы, то есть векторы прерываний, системные таблицы и области данных.

Общеизвестно, что любой отладчик забирает на себя обработку как минимум двух прерываний — первого и третьего. Проверить, не переустановлены ли эти векторы на какую-либо программу, которая скорее всего является отладчиком, можно двумя способами. Во-первых, можно проверить код первой команды обработчиков этих прерываний, который, как правило, равен 0CFh (iret). Чтобы избежать ложной тревоги в какой-нибудь лицензионной операционной системе, можно воспользоваться более надежным способом, а именно проверить, располагаются ли обработчики этих прерываний в системной области DOS. Проверка осуществляется путем сравнения сегментного адреса обработчика с сегментным адресом первого MCB и основана на том, что области памяти, в которых расположены системные обработчики, не описываются с помощью MCB. Сегментный адрес первого MCB можно получить при помощи функции 52h, которая возвращает в регистрах ES:BX указатель на блок параметров операционной системы. Со смещением -2, то есть по адресу ES:[BX-2], расположен сегментный адрес первого блока управления памятью.



Другой однозначный признак работы программы под отладчиком связан с механизмом порождения дочерних процессов в MS-DOS. Дело в том, что command.com (или любой другой командный процессор) имеет статус родительского процесса, а любая запускаемая им программа — дочернего. В случае автономного запуска программа будет иметь над собой только родительский процесс, а в случае запуска под отладчиком — дочерний процесс отладчика. Распознать, каким является процесс, запустивший программу, родительским или дочерним, можно следующим образом. В PSP дочерней программы со смещением 16h хранится сегментный адрес PSP процесса-предка. В свою очередь у процесса-предка в PSP также есть сегментный адрес PSP собственного предка, который для command.com совпадает с сегментным адресом собственного PSP. Следует отметить немаловажную деталь: это утверждение справедливо для любой копии командного процессора.

Схематично эту ситуацию можно изобразить следующим образом.



Работа без отладчика — Работа под отладчиком

После начала работы программы проверку можно осуществить с помощью несложного набора команд:

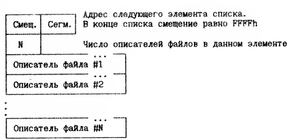
```
        mov     ax,es:[16h]     ; ES -> собственный PSP
        mov     es,ax           ; ES -> PSP процесса-предка
        cmp     ax,es:[16h]     ; У предка есть предок?
        jne     Debugger
        ...
Debugger:                       ; Процедура реакции на отладчик
```

Еще одним признаком, связанным с порождением дочерних процессов, но, правда, присущим не всем отладчикам, служит адрес завершения программы, располагающийся в PSP со смещением 0Ah. Когда операционная система загружает программу на выполнение, она копирует в это поле текущее значение вектора 22h. При загрузке программы из отладчика возможны две равноправные ситуации. Либо отладчик переустанавливает на себя 22h прерывание и затем загружает программу, либо отладчик сначала загружает программу, а затем корректирует в ее PSP адрес возврата (так, например, поступает TD). Соответственно во втором случае он будет отличаться от значения вектора 22h.

Последний признак присутствия отладчика также базируется на определении того процесса, который яв-

ляется родителем для процесса программы. В данном случае это делается другим, более изощренным способом. Как было указано выше, функция 52h возвращает в ES:BX адрес блока параметров операционной системы. Со смещением +4 в этом блоке расположен адрес первого элемента таблицы файлов. Это те самые таблицы, которые операционная система использует при обращении к файлам через handle.

Каждый элемент таблицы имеет следующую структуру:



Как правило, операционная система организует два элемента списка*. Первый рассчитан на пять, второй — на оставшееся число описателей (общее число описателей задается параметром FILES в файле config.sys).

В описателе файла, длина которого 53 байта, нас будут интересовать два поля. Первое поле имени файла. Его длина 11 байт, смещение от начала описателя 20h. Второе поле — сегментный адрес PSP задачи, открывшей файл, смещение от начала 31h.

Если программа до начала работы защитного механизма не открывала файлов, можно использовать следующий алгоритм обнаружения присутствия отладчика. Получив свое имя из окружения и преобразовав его в соответствующий формат (имя файла и расширение записываются заглавными буквами и дополняются соответственно до 11 и 3 символов пробелами), программа ищет его в таблице открытых файлов. Если имени в таблице нет, это значит, что операционная система затерла его, открывая какие-то файлы для отладчика. Если же имя программы обнаружено, то анализируется, кто открыл файл. Анализ можно проводить либо способом, описанным выше, — через поиск родителя у родителя, либо через поиск имени родительского процесса. Как уже упоминалось, окружение DOS начиная с версии 3.30 хранит путь загрузки программы, если (NB!) эта программа не является командным процессором. Для него путь загрузки в окружении отсутствует.

* Командная строка "базового" командного процессора хранится в формате ASCIIZ с некоторым смещением после первого элемента списка таблицы файлов. Если ваша программа запускалась именно "базовым" командным процессором, то командная строка будет содержать имя файла программы. Вероятность такого события из-за большой популярности оболочек очень мала, но не мешает проверить, содержит ли командная строка имя программы, и если да, то — на каком месте оно стоит.

В заключение о практическом применении описанных сведений. Вы вправе задать вопрос: если эти сведения доступны и взломщику, кто же помешает ему скрыть "следы" отладчика, скорректировав соответствующим образом области системных данных и другие элементы среды программы? Да, с одной стороны это так. Описанные методы обнаружения не являются панацеей, и против них тоже можно найти противоядие. Но с другой стороны, подобных методов обнаружения отладчика может быть достаточно много, ведь в данной статье описано лишь то, что лежало на поверхности. При их комплексном применении значительно повышается вероятность того, что взломщик не учтет либо ошибется при реализации какого-либо момента. К тому же создание корректной среды при исследовании программы с помощью отладчика должно существенно увеличить время "вскрытия". Это плюсы с точки зрения защиты. С точки зрения взлома данная информация может быть использована при создании эмуляторов программной среды.

Д.Правиков

Литература:

1. Расторгуев С.П., Дмитриевский Н.Н. Искусство защиты и "раздевания" программ. — М.: СОВМАРКЕТ, 1991.
2. Долгин А.Е. Защита программного обеспечения от исследования. // Несанкционированное копирование информации №1. — М.: Издательство агентства "Яхтсмен", 1992.
3. Спесивцев А.В., Вегнер В.А., Крутяков А.Ю., Серегин В.В., Сидоров В.А. Защита информации в персональных ЭВМ. — М.: Радио и связь, 1992. — С. 91-102.
4. Фролов А.В., Фролов Г.В. Операционная система MS-DOS. — Т. 1, кн. 3. — М.: "Диалог — МИФИ", 1992. — С. 192-195.
5. Богданов В., Де-Мондерик А., Касперский Е. Несколько советов авторам вирусов. // КомпьютерПресс, 1992, №9.
6. Dave Williams. Programmer's Technical Reference for MS-DOS and the IBM PC, 1989.

Новая версия ALDUS PAGEMAKER — что нового?

7 января этого года фирма Aldus заявила о скором начале продаж новых версий ставшего классикой издательского пакета PageMaker. Будут выпущены версии для Macintosh и для MS Windows, а свершится это в первой половине 1993 года. Обе версии будут практически идентичны. Сообщается, что в Aldus PageMaker 5.0 добавлено и усовершенствовано свыше сотни функций.

Президент фирмы Aldus Пол Брайнед рассказал, что фирма ожидает полностью окупить затраты и получить первые прибыли уже во втором или третьем квартале, причем большая часть денег поступит от нынешних пользователей пакета, которые перейдут к его новой версии. Примерно половина из них, по оценкам фирмы, сделает это в течение первого года существования PageMaker 5.0.

Начиная с 1984 года, когда появился пакет, было продано около миллиона копий, и половина покупателей зарегистрировала их.

Спикер (не подумайте чего плохого!) фирмы Пэм Миллер рассказал, что фирмы Aldus и Kodak подписали соглашение об использовании технологии управления цветом. Фирма Aldus надеется, что эта технология будет встроена в PageMaker к середине года. В новой технологии будут поддерживаться различные средства управления цветом. Aldus разработала стратегию открытой архитектуры с использованием API, способной вызывать различные цветовые системы.

Одна из целей выпуска новой версии PageMaker — догнать основного конкурента, пакет Quark Express. Фирма надеется, с помощью 5.0 вернуть кусок рынка, откушенный Quark'ом.

Одно из важнейших дополнений — возможность совместной работы с другими программами. Добавлены фильтры импорта и экспорта для Kodak Photo CD, файлов DXF, WordPerfect Graphics и для преобразования файлов Metafile и PICT. Теперь PageMaker 5.0 — как в среде Windows, так и на компьютерах Macintosh — поддерживает связь OLE (object linking and embedding) в качестве клиента, новую систему обслуживания шрифтов, которая определяет, какие шрифты из примененных в публикации недоступны, и подбирает доступные фонты (это особенно полезно при переносах публикаций между средами Windows и Macintosh).

Еще одно важное дополнение — поддержка стандарта TWAIN для прямого сканирования (см. КомпьютерПресс №8'92). Теперь пользователь может "брать" изображеиие в публикацию прямо со сканера, видеоплаты или камеры. Можно использовать EPS-файлы, имеется возможность не только импортировать их, но и привязывать к публикации, чтобы следить за их изменениями.

Интересно, что на этот раз фирма выпустила одно и то же руководство для обеих версий.

По просьбам трудящихся, трýдящихся в PageMaker, в версию для Windows добавлена управляющая панель, а в версии для Macintosh она усовершенствована. Это позволило задать точное положение и размер текста и графики на странице. Управляющая панель позволяет устанавливать параметры текста и абзацев, такие как шрифт, стили, размер, сдвиг базовой линии, отступы и выравнивание. Есть как кнопки прокрутки, так и цифровое поле для точного задания параметров.

Теперь пользователи могут увеличивать фрагменты страницы до 8 раз, использовать шрифты размером от 0.1 до 800 пунктов и отдельно задавать параметры для линий и закраски.

Представители фирмы заявили, что в новых версиях полностью переписана часть программы, генерирующая коды для PostScript-устройств, что призвано ускорить печать, в том числе фоновую (хотя, конечно же, скорость печати определяется прежде всего используемой аппаратурой и сложностью печатаемого файла). Кстати, можно печатать любые страницы в любом порядке.

Есть еще несколько новых возможностей, например, возможность задать и сохранить установки принтера, что позволяет печатать несколько файлов в пакетном режиме, причем каждый из них со своими установками. Теперь можно легко переносить текст и графику между несколькими открытыми публикациями.

Требования к системе для использования версии 5.0 для Windows: как минимум Windows 3.1, Windows-совместимый компьютер с 4 Мбайтами оперативной памяти и 80-Мбайтным винчестером, видеоадаптером высокого разрешения и мышью.

На Macintosh'е понадобится операционная система System 7 или System 6.0.7 с Finder 6.1.7, компьютер Mac II, Quadra или SE/30 с 5-8 Мбайтами оперативной памяти и винчестером емкостью 80 Мбайт.

Aldus заявила, что первой появится версия для Windows, через 30 дней начнутся поставки PageMaker 5.0 для Macintosh. Точных дат при этом не называлось — так, "где-то в первой половине года..."

Рекомендованная цена для PageMaker 5.0 составит 895 долларов. Зарегистрированные пользователи PageMaker 4.2 для Macintosh или 4.0 для Windows, купившие пакет после 1 января 1993 года, имеют право получить новую версию бесплатно. Те, кто приобрел PageMaker раньше, могут получить новую версию за 150 долларов. Исключение сделано еще для владельцев версии 4.2 для Macintosh — для них новая версия стоит только 75 долларов.

По материалам, подготовленным Джимом Маллори для Newsbytes News Network



Сегодня мы представляем фирму ELCO Technology, за короткое время ставшую одной из наиболее сильных фирм на отечественном компьютерном рынке. Чтобы рассказать о ней читателям, наш корреспондент встретился с генеральным директором фирмы Григорием Элькиным.

ELCO Technology

КомпьютерПресс: Первый вопрос — по поводу появления фирмы ELCO Technology — когда и как это произошло?

Григорий Элькин: История появления ELCO Technology очень проста. Основной состав, который работает здесь, работал в одном из совместных предприятий. Как раз началось время, когда совместные предприятия стали испытывать сложности. Коллектив был дружный, работоспособный и высококвалифицированный (по крайней мере, в области сетевых технологий). Этот коллектив надо было сохранить. И тогда основная инициативная группа приняла решение создать компанию, которая будет специализироваться в области сетей и коммуникаций. Так появилась ELCO Technology.

КП: Когда это было?

Г.Э.: Дело в том, что сама идея создания фирмы появилась летом 1991 года, а реально ELCO Technology заработала в начале 1992 года.

КП: Каковы основные области, в которых специализируется ваша фирма?

Г.Э.: Основная деятельность — это услуги в области сетей. Те специалисты, которые здесь работают, давно имеют хороший контакт с фирмой Novell. Из четырех первых в стране инженеров, сертифицированных Novell в области сетей ЭВМ, трое работают в ELCO Technology. Поэтому в основном все строится вокруг Novell. Сама фирма динамично развивается от идеологии локальных сетей к идеологии открытых сетей. Другие наши партнеры выпускают разнообразное оборудование для создания современных сетей и систем компьютерных коммуникаций.

КП: Ваша основная специализация — сети, сети для любых областей. Речь идет только о локальных сетях или о глобальных тоже?

Г.Э.: Нет, речь не идет только о локальных сетях. Дело в том, что обычно Novell знают как фирму — разработчика операционной системы для локальных сетей; очень мало знают о различных коммуникационных продуктах Novell. Это и порты для различных архитектур, начиная с X.25, TCP/IP, коммуникационные серверы, системы межсетевых коммуникаций и т.д. Множество продуктов. Сейчас ими начинают пользоваться и у нас в стране. Но пока это простейшие продукты — такие, как серверы доступа. Что касается открытой архитектуры, этого пока мало. Но я думаю, что уже в этом году о ней узнают больше. К сожалению, в этом, наверное, во многом вина дистрибьюторов Novell, которые рекламируют в основном операционные системы для локальных сетей и меньше — коммуникационные продукты.

КП: Сейчас компьютеризация страны продолжается именно в области объединения компьютеров и компьютерных систем. Здесь сети важны, но при создании таких могучих систем нужно думать о безопасности, о которой у нас, как правило, не думает никто. Скажем, у банка есть десять филиалов; понятно, что если информация будет гулять в открытом виде по любым каналам, то она будет регулярно теряться. Позволяет ли идеология Novell обеспечить безопасность при построении сетей?

Г.Э.: Безусловно. Есть целый ряд продуктов, позволяющих защищать информацию в процессе ее передачи.

КП: То есть продукты Novell можно использовать в банках, на биржах и т.п.?

Г.Э.: Тут нужно сказать вот что. У нас в стране, поскольку имеется большое количество средних и малых банков, внимание банков к продукции Novell достаточно велико. Если обратиться к мировому опыту, то приходится искренне признать, что место Novell как основных коммуникационных систем в банках, особенно в крупных, работающих в банковских сетях, весьма незначительно. Где-то, в каких-то подразделениях стоят локальные сети, но это лишь элементы сети, используемой в банке. И когда к нам обращаются большие банки, я никогда не рекомендую им Novell как основную систему. Все-таки мировой опыт много значит.

КП: Чем занимается фирма кроме сетей? Есть ли какие-то другие области деятельности?

Г.Э.: Нет. Все направлено на сети и все вокруг сетей. Мы имеем много партнеров в мире, причем работаем напрямую и достаточно успешно. Партнеров мы отбираем таким образом, чтобы решать любые вопросы, связанные с использованием се-

тей. Наши партнеры — это достаточно крупные фирмы, например, DELL Computer Corporation. Сейчас это одна из быстро развивающихся фирм. Ее компьютеры пользуются большим авторитетом на мировом рынке. Другая, также достаточно известная фирма в области компьютеров, — это AST Research. Это то, что касается серверов и рабочих станций. Мы сейчас пытаемся найти взаимные интересы и контакты с фирмами, делающими другие компьютерные архитектуры. Этим летом велись переговоры с NeXT, но NeXT, к сожалению, пока не проявляет интереса к нашему рынку. Далее, что касается сетевого оборудования, то его мы поставляем от крупных фирм, имеющих стратегические соглашения с Novell и права на поставку оборудования под маркой Novell. Мы поставляем оборудование таких фирм, как Microdyne и Eagle. Это дорогое и очень качественное оборудование, признанное во всем мире. Недавно мы заключили дистрибьюторский контракт с фирмой Cabletron. Это не очень известная фирма на нашем рынке, хотя в мире это одна из самых крупных фирм в области производства сетевого оборудования. Кроме того, в области сетевого оборудования мы сотрудничаем с компанией Compex. Это американская компания, которая производит оборудование на Сингапуре. Это оборудование недорогое, но наш опыт и опыт других российских компаний показывает, что качество его достаточно высоко. Вообще, все оборудование, о котором я говорю, сертифицировано Novell, в том числе Compex. Насколько я знаю, в России многие компании поставляют оборудование Compex.

КП: Оно дешевле и, видимо, чуть менее надежно?

Г.Э.: Да, но вполне приемлемо. И далее, что касается аксессуаров. Это различные кабели, переходники, блоки бесперебойного питания и так далее. В отношении блоков бесперебойного питания мы являемся дистрибьюторами American Power Conversion и голландской фирмы Victron. В части аксессуаров наша фирма является партнером американской фирмы Advanced Electronic Support Products. Фирма производит различные кабели, соединители и так далее, в общем все то, что называется аксессуарами, причем высокого качества. Эта фирма является официальным поставщиком IBM, изготавливает по заказу IBM кабели. Наверное, основных наших партнеров я перечислил.

КП: И еще один вопрос. Говоря о сетях, мы совершенно не вспомнили про модемы.

Г.Э.: С модемами обстоятельства несколько сложнее. Мы продолжаем искать партнера, который поставлял бы хорошие модемы именно для наших условий. Пробовали работать с некоторыми фирмами, которые имеют сертификаты Минсвязи России на свои модемы, однако этот опыт был не очень удачным.

Сейчас мы договорились об испытаниях оборудования американской компании Boca Research. Эта договоренность интересная. Boca Research согласна даже кое-что изменить в своем оборудовании, чтобы работать на наших линиях. В ближайшее время мы получим партию этих модемов и будем смотреть их в работе. Но пока, к сожалению, приходится признать, что предложить хорошие модемы для наших условий мы не можем.

КП: Следующий вопрос: чем определяется ваша политика в области цен?

Г.Э.: Во-первых, мы пытаемся объяснять своим партнерам возможности нашего рынка, и иногда нам удается убедить их, что если они хотят на этот рынок выйти, то нужно пытаться работать по каким-то более-менее приемлемым ценам. Так, например, в результате переговоров с Novell (это было еще до создания ELCO Technology) специально для Союза появилась русская версия NetWare, которая почти в два раза дешевле, чем стандартная англоязычная версия. Конечно, в основном, нам приходится придерживаться той политики цен, которую проводят фирмы. Наша политика заключается в нашей прибыли. В последнее время мы стараемся работать на небольшом проценте прибыли. Мы считаем, что следует научиться работать именно так, потому что, как только стабильности в нашей стране станет больше, сюда придут очень крупные дистрибьюторы, такие как настоящий Merisel (не тот маленький Merisel, который сейчас работает здесь). И тогда нас просто не станет, если мы не сможем работать с большим объемом и с маленькими прибылями. Поэтому основное в нашей политике цен — это работать с небольшой нормой прибыли. Например, с нашими партнерами, с реселлерами мы работаем с прибылью 4-6, максимум 10%. Конечно, когда мы работаем с конечным пользователем, прибыль составляет 10-12%, но этот процент нужно снижать. Например, в начале 1992 года мы, как принято у нас в стране, получали 20% прибыли. С ростом объема продаж снижаем этот процент.

КП: Но тем не менее, цены достаточно высоки.

Г.Э.: Цены определяются уровнем цен на это оборудование в мире.

КП: Понятно. А этот уровень, в свою очередь, определяется качеством оборудования.

Г.Э.: Безусловно.

КП: О чем еще можно сказать, говоря об ELCO?

Г.Э.: Пожалуй, нужно сказать еще о стратегии нашего маркетинга на рынке. Мы пытаемся использовать три линии маркетинга. Первая — это просто общая, то есть рассчитанная на всех, реклама в "Известиях", в компьютерных газетах, компьютерных журналах.

Вторая линия маркетинга рассчитана на специалистов. Это проведение целевых конференций. Например, в октябре мы провели "всеСНГовскую" конференцию по сетям, на которой приняли решение об учреждении Ассоциации пользователей продукции фирмы Novell. В мире существует организация, называющаяся NetWare User International. Novell тратит очень много денег на эту организацию, на ее поддержку. Например, есть бесплатный журнал, который распространяется в рамках этой организации, региональные отделения получают право на издание различных бесплатных бюллетеней, журналов и т.д. Вот такие целевые конференции мы и проводим. Сейчас готовятся конференции в Нижнем Новгороде и Петербурге.

То есть это маркетинг, рассчитанный на компьютерных специалистов, донесение до них информации о новых продуктах, а также о старых, но неизвестных на рынке. До апреля, до выставки Comtek'93, мы открываем у себя бесплатный сервер доступа для всех специалистов, с тем чтобы они имели возможность получить информацию о ценах на различную продукцию у нас и не у нас, информацию о продуктах. Будет доступна энциклопедия NetWare — это огромный объем информации обо всех, и новых, и старых продуктах Novell, о проблемах их использования. Из энциклопедии можно будет получить новые драйверы, которые бесплатно распространяются, и т.д.

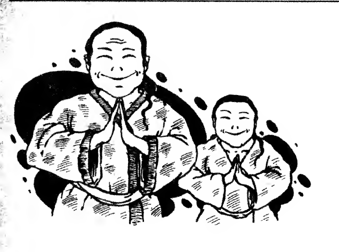
Как видите, все эти различные мероприятия связаны с работой со специалистами, так как именно через специалистов эта информация должна доходить до тех, кто инвестирует деньги.

И третье направление — это работа с теми, кто имеет деньги. Это попытки найти контакт с большими государственными и коммерческими структурами. И все эти три направления приносят определенный результат.

КП: Каким образом специалистам попасть на конференции?

Г.Э.: По опыту конференции, которая проводилась в Москве,— была еженедельная реклама в "Известиях", была реклама по радио, то есть это рекламировалось достаточно широко. В итоге в конференции приняло участие около 400-450 человек. Точно так же будет и для предстоящей конференции. Готовится рекламная кампания.

Хочу пригласить читателей КомпьютерПресс на наш стенд на выставке Comtek'93, где можно будет узнать много нового о сетях ЭВМ и получить ответы на волнующие вас вопросы.



Несмотря на то что чуть ли не каждый день появляются все новые и новые модели струйных, лазерных и термопринтеров, они, видимо, еще не скоро вытеснят с компьютерного рынка старые добрые матричные печатающие устройства, основанные на ударном принципе (так называемые “игольчатые“ принтеры). Пожалуй, вся основная работа по подготовке различных документов и текстов программ приходится именно на эти неприхотливые “рабочие лошадки“ пользователей персональных компьютеров. В этой статье мы расскажем о некоторых из них.

"Рабочие лошадки" от Epson

Все технические материалы для этой статьи были любезно предоставлены редакции КомпьютерПресс фирмой **Computer Mechanics**. Надо отметить, что из дюжины моделей матричных принтеров, выпускаемых с маркой Epson, фирма Computer Mechanics предлагает уже сегодня практически все, за исключением, пожалуй, трех-четырех наиболее дорогих моделей.

Вообще говоря, популярность “игольчатых“ принтеров (которые мы будем здесь называть и матричными, хотя это, разумеется, более широкое понятие) легко объяснима. Во-первых, для таких печатающих устройств требуются достаточно дешевые расходные материалы: обыкновенная бумага и сравнительно недорогая красящая лента. Во-вторых, большинство матричных ударных принтеров позволяют работать как с отдельными форматными листами, так и с рулонной бумагой. И, в-третьих, скорость работы и качество печати современных “игольчатых“ принтеров удовлетворяют требованиям подавляющего большинства пользователей персональных компьютеров. Недаром в “стандартный“ набор “286, 1-40, косые, мышь“ входит обычно и матричный принтер.

Хотя на отечественном компьютерном рынке в настоящее время при желании можно найти практически любую нужную (да, впрочем, и ненужную) модель матричного принтера, не слишком погрешив против истины, можно заметить, что наиболее популярными являются модели с марками Star, NEC, Oki, IBM, ну и, конечно, Epson. Многие пользователи персональных компьютеров, наверное, до сих пор применяют (или по крайней мере помнят) такую популярную модель принтера этой фирмы, как FX-80. Но, как известно, времена меняются, и современные модели принтеров Epson по техническим параметрам существенно превосходят своих недавних предшественников. Попробуем на примерах нескольких новых моделей отметить лишь самые характерные черты ударных печатающих устройств с маркой Epson.

Начнем, пожалуй, с недорогих 9-игольчатых моделей: FX-870/1170 (соответственно с узкой и широкой кареткой, 80/136 — ширина печати в символах PICA). Эти принтеры по праву соответствуют образу идеального на сегодняшний день принтера — типа “рабочая лошадь“. Их конструкция рассчитана на тяжелую ра-

боту при высокой производительности — так называемый стиль havy duty design. Модели FX-870/1170 в режиме Draft ("черновой" печати) работают просто с изумительной скоростью — 380 символов в секунду. Это на 30% быстрее, чем допускалось в предыдущих моделях FX-850/1050. В режиме NLQ (Near Letter Quality) на каждый печатаемый символ отводится 57 точек. Кстати, сменный картридж с красящей лентой рассчитан на печать 3 миллионов символов.

Новые модели принтеров гарантируют надежную подачу бумаги практически любого типа, включая отдельные форматные листы, бумажную ленту, многоэкземплярные формы, конверты, этикетки и т.п. Дополнительное устройство протяжки дает возможность загружать два типа бумажной ленты. Отдельные форматные листы можно загружать автоматически сверху или спереди, используя дополнительные загрузчики бумаги. Большое удобство для работы дает такая функция этих моделей принтеров, как Paper Park (парковка бумаги), обеспечивающая комбинированную подачу используемой бумаги. В этом случае бумажная лента подается в устройство протяжки сзади и находится там в состоянии ожидания до тех пор, пока вы работаете с другим типом бумаги (например, с форматными листами).

Немаловажное значение при работе с вышеупомянутыми моделями принтеров имеет и возможность точного выравнивания бумаги. Это, в частности, позволяет подавать бумагу вперед и назад на крошечные интервалы (1/180 дюйма), например для точного размещения некоторых записей при печати. При работе с бумажной лентой позиция загрузки может быть также отрегулирована достаточно точно и запомнена так, что устройство протяжки всегда будет возвращать бумагу в одно и то же положение.

Загрузчик листов с одним приемником бумаги способен хранить до 50, а высокопроизводительный загрузчик — до 150 листов бумаги. Таким образом, при совместном использовании обоих загрузчиков появляется возможность хранить готовыми к печати до 200 отдельных форматных листов.

Кстати, еще одной интересной особенностью этих моделей принтеров является возможность получения

Таблица 1

Модель принтера Epson	Количество игл	Скорость печати, знак/с	Разрешающая способность точек/дюйм	Ширина печати знаков	Код управления Epson	Виды шрифтов
LX-100	9	240	240	80	ESC/P	Draft, Roman, Sans Serif
LX-1050	9	290	240	136	ESC/P	Draft, Roman, Sans Serif
FX-870	9	380	360	80	ESC/P	Draft, Roman, Sans Serif
FX-1170	9	380	360	136	ESC/P	Draft, Roman, Sans Serif
LQ-100	24	200	360	80	ESC/P 2	Draft, Roman, Sans Serif Courier, Prestige, Script
LQ-870	24	225	360	80	ESC/P 2	Draft, Roman, Sans Serif Courier, Prestige, Script OCR-B, Script-C, Orator(S)
LQ-1170	24	225	360	136	ESC/P 2	Draft, Roman, Sans Serif Courier, Prestige, Script OCR-B, Script-C, Orator(S)

трех бескопирочных (то есть без использования "копирки") копий. "Иголки" в состоянии пропечатывать эти копии через оригинал.

Данные модели принтеров используют, как обычно, стандартный управляющий код Epson ESC/P. Из 13 международных наборов символов, поддерживаемых данными моделями принтеров, наибольший интерес для отечественных пользователей, безусловно, представляют наборы 866 (альтернативная кодировка) и GOST USSR (основная кодировка).

Другие новые модели принтеров — LQ-870/1170 (с узкой и широкой кареткой) — используют уже 24-игольчатую печатающую головку, благодаря чему и обеспечивают практически типографское качество печати — LQ (Letter Quality). В этом классе принтеров, пожалуй, нет других моделей, которые могли бы состязаться с их скоростью печати чернового шрифта (режим Draft) — до 275 знаков в секунду. Для высококачественной печати (режим LQ) скорость печати не превышает 92 знаков в секунду.

Для этих моделей принтеров используется уже новый международный управляющий код Epson ESC/P 2. Для пользователя этот код обеспечивает, по крайней мере, два неоспоримых преимущества: во-первых, управление размером шрифта и, во-вторых, расширенный графический режим. В новом управляющем программном обеспечении расширена и улучшена поддержка шрифтов национальных языков, а возможность управления ими во время печати приближает принтеры, использующие коды ESC/P 2, к возможностям дорогих страничных печатающих устройств. Встроенные шрифты Roman и Sans Serif могут масштабироваться от 8 до 32 пунктов с приращением в 2 пункта. Кроме этого, предусмотрена также возможность масштабирования шрифтов, которые еще появятся на компьютерном рынке.

Улучшенный графический режим позволяет печатать с разрешением 360 точек на дюйм, как по горизонтали, так и по вертикали (то есть в обоих направлениях). Понятно, что в таком режиме обеспечивается просто превосходное качество графического изображения. Напомним лишь, что сравнительно недорогие модели лазерных принтеров обеспечивают разрешающую способность всего 300 точек на дюйм.

В отличие от 9-игольчатых принтеров, модели LQ-870/1170 обеспечивают уже не менее пяти различных трактов подачи бумаги: два для форматных листов и три для бумажной ленты. Функциональная гибкость новых моделей принтеров в отношении подачи бумаги заключается, в частности, в возможности быстрого снятия и перестановки трактора (узла подачи бумаги). Трактор может быть установлен в положение подачи сзади или спереди или в положение протяжки сверху, в зависимости от того, какое из них наиболее подходит. Форматные листы могут подаваться сверху и спереди, а бумажная лента — сзади, спереди и снизу. Кстати, любая печать на форматных листах сопровождается автоматическим складированием готовых экземпляров. Отдельные характеристики некоторых современных моделей принтеров Epson приведены в табл. 1.

Для более подробного ознакомления редакции журнала фирмой Computer Mechanics была предоставлена новая модель принтера Epson LQ-100. Эта модель действительно заслуживает того, чтобы рассказать о ней более подробно. Пожалуй, она вобрала в себя многие характерные особенности, присущие современным моделям принтеров с маркой Epson:

- скорость печати в режиме Draft достигает 200 символов в секунду;
- используется широкий набор наиболее популярных шрифтов, в том числе и масштабируемых;
- поддерживается новый международный управляющий код ESC/P 2, а также расширенный графический режим (360 точек на дюйм);
- сохранена совместимость с управляющим кодом ESC/P, по-прежнему применяемому в других моделях принтеров Epson;
- допускается два рабочих положения принтера (вертикальное и горизонтальное) и комбинированная подача бумаги;
- предельно упрощено управление принтером (двухкнопочная панель).

Таблица 2

Шрифт	10 cpi	12 cpi	15 cpi	Пропорциональный
Draft	+	+	+	–
Roman	+	+	+	+
Sans Serif	+	+	+	+
Courier	+	+	+	–
Prestige	+	+	–	–
Script	+	+	–	–

Начнем с того, что габаритные размеры новой модели составляют 130 на 375 на 252 мм (высота)(ширина)(глубина), а вес не превышает 4,5 кг. Данный принтер осуществляет матричный принцип печати при помощи 24 иголок. Максимальная ширина печатной строки (при размере символов 10 cpi — знаков на дюйм) не превышает 80 знаков или 137 знаков в режиме Condensed. Срок службы сменного картриджа с красящей лентой рассчитан на печать около 2 миллионов символов, причем это в случае, если на каждый символ отводится 48 точек. Каждая игла печатающей головки за время работы должна выполнить не менее 100 миллионов ударов.

Отдельные листы для печати должны иметь либо американский формат (8,5 на 11 дюймов), либо формат А4 (210 на 297 мм). Модель LQ-100 оснащена загрузчиком листов, который может хранить не менее 50 таких форматных листов. Кстати, модель допускает и ручную подачу отдельных листов. При использовании бумажной ленты (шириной до 10 дюймов) можно, помимо печатаемого оригинала, получать и одну копию. Возможна также печать конвертов и этикеток. Заметим, что для печати отдельных форматных листов и конвертов используется фрикционная подача, а для бумажной ленты и этикеток — специальный трактор. Немалое удобство при работе с данной моделью дает уже упомянутая функция Paper Park. Механизм подачи бумаги может обеспечивать выравнивание с точностью 1/360 дюйма.

Надо сказать, что работа с данной моделью принтера доставляет истинное наслаждение. Особенно привлекательной для пользователя может оказаться возможность вертикального размещения принтера — ведь места на столе, где установлен компьютер, как правило, катастрофически не хватает. Используя всего две кнопки (FONTS/YES и FORM FEED/NO), можно выбрать один из следующих режимов: тестирование, демонстрация, выбор шрифтов (см. табл. 2) и их масштабирование, ручной режим загрузки бумаги.

Начальные установки, вводимые на старых моделях принтеров при помощи микропереключателей, производятся для модели LQ-100 в режиме диалога. После включения принтера (одновременно с нажатой клавишей YES) пользователю задается (печатается на бумаге) ряд вопросов, на которые он просто должен ответить "да" или "нет", нажав соответственно клавишу YES или NO. Кстати, первым является вопрос о выборе языка диалога. Предлагаются, в частности, немецкий, итальянский, испанский и, конечно, английский. На этом же этапе пользователь должен выбрать, например, используемую кодовую таблицу, формат бумаги, направление печати (в зависимости от расположения принтера), возможность загрузки шрифта пользователя и т.п.

В частности, пользователь может выбрать один из 15 международных наборов символов, включая наборы 866 и GOST USSR. Надо отметить, что направление печати в графическом режиме может выполняться как в одном, так и в обоих направлениях. Эту опцию пользователь также должен выбрать на этапе начальных установок. Как правило, для воспроизведения наиболее качественных изображений используют однонаправленную печать, хотя понятно, что она и медленнее. Для загрузки шрифта пользователя во входном буфере размером 11 Кбайт выделяется около 9 Кбайт памяти.

Для связи с компьютером принтер, как обычно, оснащен стандартным интерфейсом Centronics. В комплект с принтером входит его описание, достаточно ясное и подробное. В него включены, в частности, такие разделы, как подготовка принтера к работе, выполнение начальных установок, действия пользователя при аварийных ситуациях и т.д.

Все интересующие вас вопросы вы можете задать специалистам фирмы Computer Mechanics, позвонив по *телефонам (095) 939-32-66, 939-59-77*. Вас ждут по адресу: *Москва, Мичуринский проспект, д.1*.

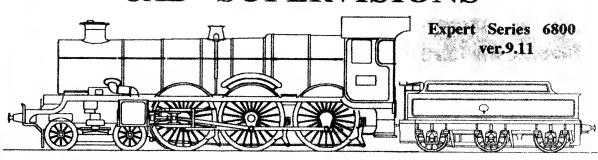
А.Борзенко



мастер-дилер фирмы ATS

ПРЕДСТАВЛЯЕТ НОВУЮ ВЕРСИЮ

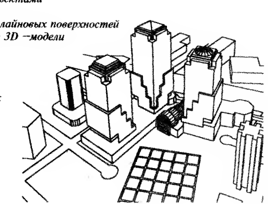
CAD SUPERVISIONS™



Реальное 3D-моделирование и конструирование.
Неограниченные возможности применения в машиностроении, архитектуре, дизайне.

Пакет **SUPERVISIONS** включает в себя:

- *простой и доступный интерфейс пользователя*
- *встроенный редактор текстов*
- *встроенный научный калькулятор*
- *макроязык SPL*
- *параметризацию моделей*
- *булевы операции с трехмерными обьектами*
- *большой набор сплайновых кривых*
- *возможность создания сложных сплайновых поверхностей*
- *получение двухмерных чертежей из 3D —модели*

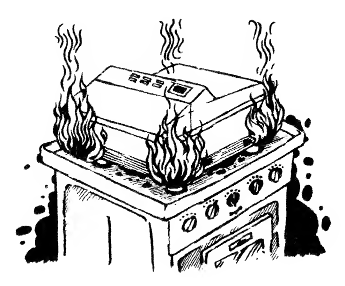


- *оформление чертежей в соответствии с ЕСКД*
- *расчет геометрических и массо—инерционных характеристик*
- *быстрый алгоритм цветового тонирования и подавления невидимых линий*
- *возможность создания слайд—фильмов*
- *открытый интерфейс*
- *преобразование данных в форматы IGES—4.0, DXF, MOSS, ISIF, Ventura, PageMaker*

Стоимость лицензионного пакета **SUPERVISIONS 6800 ver.9.11** 238 тыс. рублей, включая НДС. Возможна поставка в кредит с рассрочкой на 5 месяцев. Предприятиям, приобретающим 50 пакетов и более, МП "САЛМА" обеспечивает 5-ти дневную бесплатную стажировку одного представителя в учебном центре ATS Training Centre (Сингапур).
Предлагаем оргтехнику, программно-аппаратные комплексы любой конфигурации.

МП "САЛМА" Москва, Зеленый проспект, д.2/19 *тел. (095) 306—00—26, 306—47—89*
факс (095) 306—02—83

Мы ждем Вас в нашем демонстрационном зале !



Мы продолжаем рассказывать о принципах работы различных типов печатающих устройств. Эта статья посвящена большой группе принтеров, печатающие системы которых используют нагревательные элементы. Именно поэтому ключевым словом для таких принтеров, как правило, является слово "термо". Чаще всего эти печатающие устройства подразделяют на термопринтеры, принтеры с термопереносом и принтеры с термосублимацией.

Принтеры с "подогревом"

Свой рассказ мы начнем, конечно, с обычных термопринтеров, более распространенное название которых у нас было (да, пожалуй, и есть) — термопечати. В нашу страну эти устройства пришли, как обычно, с Дальнего Запада, однако автору этих строк еще на заре трудовой деятельности пришлось столкнуться и с отечественными образцами подобных изделий. Но обо всем по порядку.

Вообще говоря, термопечать правильнее было бы называть термографическим печатающим устройством или сокращенно — ТПУ. Но, честно говоря, наши "доморощенные" аббревиатуры многих (в том числе и меня) всегда раздражают своей непонятностью и многовариантностью (например, при обозначении контактов разъемов), поэтому благосклонный читатель, надеюсь, простит немного технического слэнга при изложении данного материала, хотя это, может быть, "грубо и не по-европейски". Итак!

Изображение в термопринтере создается путем сконцентрированного в нужном месте локального нагревания бумаги. Понятно, что бумага в данном случае нужна не простая и даже не очень простая. Она должна иметь очень тонкое термочувствительное покрытие с двумя раздельными бесцветными компонентами. Никого, пожалуй, не удивит, что именно такая бумага называется термобумагой. При локальном нагревании цветоформирователь (первый компонент), смешиваясь с предварительно обесцвеченным красителем (второй компонент), образует видимое на бумаге пятнышко. Вообще говоря, в данном случае возможна печать различными цветами, но чаще всего (это дело химиков) используются только голубой и черный. Как правило, бумага, дающая при печати изображение черного цвета, требует более высоких температур и давления печатающей головки. Но на бумаге же, обеспечивающей печать голубого цвета, хотя она и допускает обычно более высокие скорости вывода, полученное изображение со временем бледнеет и становится непригодным для практического использования. Тут хотелось бы вспомнить одну поучительную историю, случившуюся с автором этих строк, когда он безрассудно понадеялся на достижения отечественных и мировых "термопечатников". Поправившийся текст полезной программы (еще для микроЭВМ "Электроника-60") был выведен на грязно-синюю (это вовсе не литературная гипербола) термобумагу через "родной" термо-"монстрик". Каково же было его удивление, когда через несколько месяцев на этой бумаге он не обнаружил ничего кроме редких голубых пятнышек. Так сказать, криптография наоборот. Но вернемся к теме нашего разговора.

Основными составными частями печатающей головки термопринтера являются несколько крошечных нагревательных элементов, которые расположены примерно так же, как расположены иглы в обычном мат-

ричном принтере: друг над другом в два ряда. Наиболее часто печатающие головки используют два ряда, немного смещенных относительно друг друга, по 12 нагревательных элементов в каждом. Нагрев элементов осуществляется путем пропускания через них электрического тока определенной величины. Поскольку сами термоэлементы очень маленькие, то и печатающая головка принтера, как правило, имеет небольшие габаритные размеры (всего несколько миллиметров в толщину).

Как у "игольчатых" и струйных принтеров, печатающая головка термопринтера позиционируется только в горизонтальном направлении, а подача бумаги осуществляется в вертикальном (последовательные принтеры). Не надо, видимо, пояснять, что термопринтеры относятся к группе матричных устройств. Поскольку, вообще говоря, между печатающей головкой и бумагой (подчеркиваю, самой бумагой) механический контакт отсутствует, то термопринтеры — это "безударные" устройства. К основным недостаткам термопечатей следует отнести необходимость использования специальной (и, естественно, дорогой) бумаги, которая, как уже известно, может преподнести вам ненужные "сюрпризы".

Относительно недавно (по некоторым данным, в 1982 году) появился новый класс термопринтеров, печатающих уже на обыкновенной бумаге. Такие устройства носят сейчас название принтеры с термопереносом или термографические печатающие устройства с подачей красящего вещества. Принцип действия подобных принтеров приведен на рисунке. Как и для обычного термопринтера, имеется печатающая головка с нагревательными элементами, правда, эти элементы стали еще меньше. Термопластичное красящее вещество, нанесенное на тонкой подложке, попадает на бумагу именно в том месте, где элементами печатающей головки обеспечивается должный нагрев. Конструктивно такой способ печати достаточно прост, к тому же он обеспечивает практически бесшумную работу. Достаточно часто принтеры с термопереносом имеют печатающую головку шириной максимально допустимого размера бумаги, используемой в таком принтере (так называемые строчные или страничные устройства). Лента с красителем в этом случае, разумеется, должна иметь соответствующую ширину. Понятно, что при такой ситуации отпадает потребность в механике, служащей для перемещения печатающей головки вдоль бумаги.

Еще один очень интересный метод печати используется в так называемых принтерах с термосублимацией. Хотя, возможно, термин "термосублимация" не очень удачен, но он достаточно четко поясняет, каким образом красящему веществу передается необходимая порция энергии сублимации. Теперь поясним для тех, кто уже (или еще) не помнит, что это такое. Под сублимацией понимают переход вещества из твердого состояния в газообразное, минуя стадию жидкости (например, кристаллы йода сублимируют при нагревании). Иначе этот процесс, по-моему, называли еще "возгонкой".

В таких типах принтеров имеется возможность точного определения необходимого количества красителя, переносимого на бумагу. Это удается только благодаря электронному управлению процессом испарения красящего вещества. Комбинацией цветов красителей можно подобрать практически любую цветовую палитру. Принтеры с термосублимацией используются обычно только как цветные, поскольку, пожалуй, ни один тип печатающих устройств не обеспечивает лучшей цветопередачи.

Принтеры с термопереносом и термосублимацией относятся к группе матричных "безударных" печатающих устройств. Понятно, что практически все преимущества матричных принтеров автоматически можно распространить и на эти типы принтеров. Как правило, не представляет проблемы создание практически любых типов шрифтов с различными атрибутами, нет проблем и при работе в графическом режиме. Кстати, достаточно просто реализуется здесь и негатив изображения (например, белое на черном). По четкости и резкости изображения некоторые модели таких принтеров могут поспорить с хорошими лазерными устройствами. Их разрешающая способность составляет обычно 300 dpi (точек на дюйм). По скорости печати принтеры с термопереносом, как правило, уступают современным моделям чернильных струйных печатающих устройств. Физические ограничения по скорости обусловлены в основном задержкой нагрева термоэлементов и процессом переноса красителя.

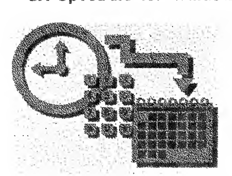
Благодаря не очень сложной механике при хорошем качестве получаемого изображения принтеры с термопереносом находят сейчас достаточно широкое применение, например в области портативных периферийных устройств для компьютеров-блокнотов. Наиболее яркими примерами таких устройств могут служить принтеры модели PN48 фирмы Citizen и модели Lexmark фирмы IBM.

А.Борзенко



Фейерверк программных продуктов

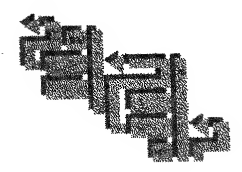
CA-UpToDate for Windows



Средство для управления ресурсами, ведения индивидуальных календарных графиков и групповых расписаний.

- **Увеличение производительности труда администратора.** CA-UpToDate позволяет управлять работой одновременно сотен людей, отслеживать необходимые события, распределять материалы, составлять графики использования помещений и т.д.
- **Оптимальное распределение времени и ресурсов.** Вы можете мгновенно находить общие временные "окна" сразу в нескольких календарных графиках, с тем чтобы назначать встречи с сотрудниками. В любой момент вы можете видеть, какие ресурсы задействованы, а какие свободны. Взглянув на экран, вы легко запланируете, организуете, проконтролируете любую встречу или событие.
- **Развитые возможности планирования.** Вы можете отслеживать более 65 000 календарных графиков. Подробные отчеты о времени, событиях и ресурсах.
- **Полный контроль над индивидуальными и групповыми расписаниями.** Просмотр, отмена, восстановление, перенос запланированных мероприятий. В любой момент можно посмотреть, кто, когда и сколько времени затратил на выполнение той или иной работы, как идет работа над интересующим проектом, кто, когда и где встречается с вашим клиентом.
- **Мгновенное восстановление прошедших событий.** Вы можете легко восстановить все свои старые контакты, встречи, переговоры, телефонные разговоры, заказы, выплаты и т.д.
- **Трехуровневая парольная защита.** Полная конфиденциальность информации. По вашему желанию информация попадет только тому, кому нужно и тогда, когда нужно.
- **"Горячие зоны"** обеспечивают сводный просмотр многих параметров.
- **Простота в использовании.** Интуитивный дизайн, контекстно-чувствительная помощь, использование всех преимуществ Microsoft Windows. Нужно научиться только основным приемам составления расписаний. После этого все, что вы захотите узнать, тут же появится на экране.
- **Работа в любых сетях, поддерживаемых Microsoft Windows.** Используйте CA-UpToDate в сети, и вы всегда будете иметь самую свежую и полную информацию о работе вашей фирмы или подразделения.

CA-SuperProject for DOS, Windows



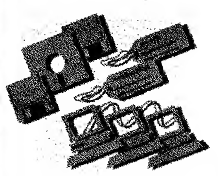
Создание и ведение разнообразных проектов в соответствии с поставленными целями, располагаемыми временем и ресурсами. CA-SuperProject предлагает проверенные средства управления проектами на основе методов сетевого планирования.

- **Простота в использовании.** Непосредственное создание проекта в графическом представлении на экране. Дружественный интерфейс, контекстно-чувствительная помощь, полная поддержка мыши. Панель быстрого доступа к типовым операциям (ToolBar), предварительная настройка "грубого" или "тонкого" режимов управления, контекстно-чувствительная обучающая система.
- **Стандартизованное управление проектами.**
- **Гибкое планирование.** Детальное описание задач проекта, сроков их

выполнения, ресурсов и затрат. Планирование работ по минутам, часам, дням, неделям, месяцам, годам. Планирование ресурсов/затрат по задачам с учетом их приоритетов, ограничений на ресурсы. Анализ реализуемости проекта при наступлении тех или иных событий. Полный контроль за ходом выполнения работ, оперативное управление проектом — корректировка графика работ с учетом меняющихся обстоятельств.

- **Управление ресурсами.** Оптимальное распределение ресурсов. Ресурсное календарное планирование. Средства выравнивания загрузки ресурсов. Распределение ресурсов между несколькими проектами.
- **Расчет затрат на всех этапах.** Системный критерий контроля стоимость/план проекта (C/SCSC). Учет переменных, постоянных и общих затрат. Полный расчет затрат по временам и задачам.
- **Разнообразные отчеты и диаграммы.** Высококачественная графика для отображения диаграмм Гантта, сетевых диаграмм (PERT), структур разделения работ (WBS), списков ресурсов/затрат. Сортировка и фильтры отбора, отчеты с различной степенью детализации данных. Поддержка всевозможных шрифтов и устройств вывода.
- **Поддержка разных уровней безопасности проекта.**
- **Импорт/экспорт данных различных форматов.** Использование данных SuperCalc, Lotus, dBASE III, CSV, Fixed ASCII, SYLK.
- **Гибкое управление памятью.** Использование extended-памяти.
- **Работа в сетях** Novell, 3Com, IBM Token Ring, Banyan и др.
- **Версии для MS-DOS, Windows, Digital VAX/VMS.** Полная совместимость файлов проектов.
- **Существует русская версия.**

CA-NETMAN/PC for DOS



Средство для менеджеров нижнего уровня, предназначенное для управления, поддержки и анализа компьютерных ресурсов конечных пользователей. Доступ к информации по любым вопросам использования компьютеров в вашей организации, оптимальное распределение оборудования, сокращение затрат на техническое перевооружение.

- **Инвентарный учет.** Учет аппаратного и программного обеспечения, офисного оборудования и пр. Учет по отдельным компонентам или по системным конфигурациям. Управление подключением систем к сетям. Легкая переброска компонентов из одной конфигурации в другую. Учет затрат на аренду и поддержку оборудования. Многоуровневый инвентарный учет конфигураций для локальных и глобальных сетей.
- **Организация эффективной службы поддержки.** Регистрация, распределение, установка очередности и слежение за своевременным выполнением заказов пользователей. Ведение учета ремонтов и профилактики по компонентам, системам и пользователям. Поиск применявшихся решений проблем по ключевым словам. Выравнивание загруженности службы поддержки, ведение учета затрат на ремонт и обслуживание.
- **Покупка и получение оборудования.** Изготовление заказов на покупку. Регистрация полученного оборудования. Подсчет затрат на приобретение. Регистрация и учет выполнения заказов пользователей на новое оборудование.
- **Гарантийное и послегарантийное обслуживание.** Учет дат начала и конца гарантийных сроков. Учет договоров и затрат на обслуживание.
- **Обучение и техническая поддержка.** Учет существующих систем обучения по продуктам. Сбор дополнительной информации по использованию продуктов. Создание внутренних справочников по обучению/использованию продуктов. Анализ затрат на обучение/поддержку.
- **Развитые средства анализа.** Анализ затрат по категориям, подразделениям, продавцам, производителям и пр. Анализ затрат по подразделениям за прошедшие периоды. Анализ бюджета подразделения в сравнении с действительными затратами. Анализ эффективности работы обслуживающего персонала.
- **Разнообразные отчеты в удобной для пользователя форме.**
- **Работа в локальных сетях.** Совместимость со всеми сетями, поддерживающими NetBIOS, а также Novell IPX/SPX.
- **Однопользовательская и многопользовательская версии.**



10 февраля 1993 года из Москвы стартовал бизнес-тур по городам Сибири и Казахстана, организованный компанией "Merisel-Компьютерные технологии". Вместе с представителями компании Merisel в туре принимали участие наиболее известные из деловых партнеров фирмы — производители средств вычислительной техники и программного обеспечения. Среди участников тура были представлены такие известные фирмы, как Intel, Seiko-Epson, Seagate, 3COM, APC, Colorado Memory Systems и Enable. Организаторы поездки пригласили в сию представительную компанию корреспондента КомпьютерПресс.

Мерисел: десант в Сибирь

Программа бизнес-тура, проводившегося в Новосибирске, Новокузнецке и Алма-Ате, состояла из пресс-конференций, презентаций фирм-участниц, показа предлагаемых аппаратных и программных средств и встреч с потенциальными партнерами. Основной целью данного тура, этой уникальной маркетинговой программы, являлся поиск новых дилеров, расширение рынков сбыта и установление надежных партнерских отношений с региональными коммерческими структурами.

"Главный успех нашей компании,— говорит генеральный директор Merisel Михаил Краснов,— состоит в том, что мы делаем ставку на лидеров. Мы предлагаем самую лучшую продукцию более 700 фирм-производителей. Главное — это не цена закупки, а цена владения. Покупая качественное оборудование и программное обеспечение, вы выигрываете во всех отношениях".

Компания "Merisel-Компьютерные технологии" — это совместное предприятие при участии американской корпорации Merisel, одного из ведущих дистрибьюторов микрокомпьютерной техники и программного обеспечения. Компания Merisel специализируется на продажах через дилерскую сеть, куда входят оптовые магазины, внедренческие фирмы и торговые дома. За два с половиной года деятельности на рынке СНГ компанией было заключено более 500 контрактов на поставку вычислительной техники.

В бизнес-туре компания Merisel помимо генерального директора была представлена Мариной Никитиной — исполнительным директором, Сергеем Эскиным — начальником отдела продаж, Александром Крыловым — сотрудником отдела продаж и рядом других ведущих сотрудников компании.

По два полных дня в каждом городе отводилось пресс-конференциям, презентациям фирм и показу образцов продукции. В Новосибирске выставка проводилась в холле гостиницы "Сибирь" и в Доме Ученых в Академгородке. В Новокузнецке мероприятия проходили в Доме Творческих Союзов, Алма-Ата предоставила для участников тура Дом Ученых АН Казахстана.

Интерес к мероприятиям бизнес-тура сильно зависел от потенциала тех городов, в которых побывали его участники. Если, например, в Новосибирске, выставку посетило около одной тысячи человек, то в Новокузнецке число посетителей достигло трех тысяч. Общее число посетителей во всех трех городах было порядка 10 000 человек, причем большинство из посетителей не представляли потенциального интереса для организаторов. По нашим оценкам, Новосибирск является наиболее компьютеризированным регионом (порядка 30-50 тыс. компьютеров при населении в 1,5 млн. человек). Новокузнецкий регион, несмотря на наличие трех крупных промышленных предприятий, представляется более благодатной почвой для компьютериза-

ции — не более 5000 компьютеров в настоящее время. Кузнецкий металлургический комбинат (КМК) имеет АСУ на базе Oracle под управлением ОС Xenix, работающую на IBM PC и миникомпьютерах Olivetti под управлением ОС RSX. В цехах расположены миникомпьютеры СМ-1 (успешно работающие на открытом воздухе). Отметим, что компания Merisel уже имела контракты с шахтой имени Ленина (г. Новокузнецк).

Лидеры

Компания Merisel привезла с собой представителей нескольких наиболее известных из своих партнеров. К сожалению, в отличие от ярко представленных производителей компьютерного оборудования, программное обеспечение было представлено лишь одной фирмой (изначально планировалось участие фирмы Borland), что несколько сдвинуло акценты в сторону фирм — производителей аппаратуры.

Итак, познакомимся с теми фирмами, которые приняли участие в бизнес-туре Merisel.

Фирма **Intel** (Владимир Нестеров — торговый представитель фирмы в Москве) — это компания номер один в мировой полупроводниковой индустрии, годовой оборот которой составил в 1992 году более 5480 млн. долларов. Потребителями продукции Intel являются такие известные фирмы, как IBM, DEC, Compaq, Olivetti и многие другие. Менее известен тот факт, что помимо изготовления микропроцессоров фирма Intel занимается разработкой сетевых адаптеров, факс-модемов, а также системных блоков на базе процессора 486, с возможностью замены на процессор пятого поколения Pentium, который планируется к выпуску во втором квартале 1993 года.

Москва, ул. Кременчугская, 6/7.
Тел.: (095) 443-97-85, факс: (095) 445-94-20.

Фирма **Seiko-Epson** (г-н Касаи (Casai) — представитель фирмы в Москве) — известный производитель матричных и лазерных принтеров, часов, промышленных роботов и оптических приборов. В ходе тура были представлены 8 новых моделей принтеров, в том числе модели со встроенными податчиками бумаги. Матричные принтеры являются наиболее дешевыми устройствами для получения нескольких копий документов. Принтеры этого типа еще несколько лет будут составлять определенную долю рынка.

Москва, Краснопресненская наб., 12,
ЦМТ, офис 1808. Тел.: (095) 253-16-79.

3COM (Найджел Мултон (Nigel Moulton) — сотрудник представительства фирмы в Великобритании) — независимая компания по производству адаптеров для локальных сетей. Планируемый оборот на 1993 год — 650 млн. долларов. Поиск продавцов на рынках СНГ. Фирма 3COM выпускает 4 основных типа сетевого оборудования: адаптеры Ethernet (1-е место в мире по производству, 32% рынка), концентраторы (3-е место в мире, 1-е в Европе), терминальные серверы (2-е место в мире) и средства для объединения разнородных сетей. Интересы фирмы представляет компания Merisel.

Фирма **Colorado Memory Systems** (Анатолий Рубанюк — представитель фирмы) является известным производителем накопителей на лентах — стримеров. Продукция фирмы включает большое количество стримеров: от 120- и 250-Мбайтных моделей Jumbo до 2-Гбайтной модели PowerTape и модели Jumbo Trakker, подключаемой к параллельному порту компьютера. В настоящее время фирма Colorado Memory Systems входит в состав Hewlett-Packard. Интересы фирмы представляет компания Merisel.

Созданная в 1979 году фирма **Seagate** (Аластер Хантер (Alastair Hunter) — региональный директор фирмы по Восточной Европе) производит 30 тыс. жестких дисков в день и входит в число 200 крупнейших компаний мира. Продукция фирмы включает различные типы жестких дисков — от 20-Мбайтных моделей для компьютеров класса PC/XT до 2,5-Гбайтных моделей, а также жестких дисков для миникомпьютеров. Недавно фирмой Seagate была приобретена фирма SunDisk — производитель флэш-памяти. Одним из дистрибьюторов продукции фирмы Seagate является компания Merisel.

Фирма **American Power Conversion (APC)** (Грэм Ингхэм (Graeme Ingham) — менеджер по Восточной Европе) является на сегодняшний день мировым лидером по производству источников бесперебойного питания (UPS) для персональных компьютеров и локальных сетей. Продукция фирмы — это две основные линии, SmartUPS и BackUPS, мощностью от 250 ВА до 2 кВА, рассчитанные на персональные и миникомпьютеры. Различные интерфейсные устройства позволяют осуществлять взаимодействия UPS с различными сетевыми операционными системами, от Novell Netware до Banyan VINES.

Москва, 109280, а/я 102.
Тел.: (095) 415-01-02.

Продукция фирмы **Enable** (Кен Норбери (Kenneth Norbery) — представитель европейского отделения фирмы) — интегрированные пакеты Enable версии 4.5, Enable Works Plus и коммуникационный пакет Huggins. Несмотря на довольно высокую цену (795 долл.), пакет Enable значительно превосходит по характеристикам своего ближайшего соперника — Framework IV фирмы Borland International. Одно из

наиболее важных достоинств Enable — это возможность работать на компьютерах класса PC/XT с памятью 512 Кбайт. Пакет Enable 4.5 включает такие компоненты, как трехмерная электронная таблица, реляционная база данных с процедурным языком, текстовый процессор, деловая графика и коммуникации. Более 500 тыс. копий было продано правительственным организациям США, Министерству обороны и Министерству финансов, совместно с компьютерами фирм Zenith и IBM. Заключено предварительное соглашение о локализации продукции фирмы Enable, которая будет выполняться компанией Merisel.

Интересы фирмы Enable в странах СНГ представляет компания Merisel.

Итоги

По окончании тура нами был проведен опрос участников с просьбой оценить по 5-балльной системе потенциал каждого города, потенциал посетителей, а также оценить перспективность регионов для фирм-участниц. Результаты выглядят следующим образом:

	Новосибирск	Новокузнецк	Алма-Ата
Потенциал города	3,4	3,2	4,6
Потенциал посетителей	3,2	3,5	3,6
Перспективность региона	3,8	3,5	4,2

Очевидное лидерство Алма-Аты объясняется в первую очередь тем, что Казахстан — это молодое независимое государство, с которым проще налаживать деловые отношения, чем с государственными или коммерческими структурами, уже имеющими связи в России. Приведем ряд комментариев, сделанных представителями фирм — участниц тура.

Enable

Представитель фирмы Enable отметил большой интерес к продукции фирмы, указав при этом, что наибольшей проблемой на сегодняшний день является соблюдение закона об авторских правах и, как следствие — борьба с незаконным копированием программного обеспечения, что, на его взгляд, более достижимо в республике Казахстан. Новосибирск обладает большим программным потенциалом и больше заинтересован в экспорте программного обеспечения собственной разработки на Запад, чем в покупке импортного программного обеспечения, пусть даже локализованного.

Intel

Фирма Intel уже имела дистрибьюторов в Новосибирске, Новокузнецк не представляет интереса из-за отсутствия специалистов, а самым перспективным городом является Алма-Ата, не только как новое самостоятельное государство, но и как регион, напоминающий Москву 3-4-летней давности по качеству и составу компьютерной техники.

Seagate

Представитель фирмы Seagate отметил, что Новокузнецк не представляет интереса для распространения продукции фирмы (по крайней мере, в ближайшее время).

Epson

Наибольший интерес для фирмы Epson может представлять не только расширение сервисной сети (сервис-центр в городе Новосибирске существует уже более года), но и участие в автоматизации промышленных предприятий (преимущественно в Кемеровской области), особенно после их приватизации.

3COM

3COM — наиболее оптимистичная фирма, присутствовавшая в туре. Трудно найти объяснение такому оптимизму, но хочется верить, что в ближайшем будущем основная масса сетевого оборудования будет от 3COM.

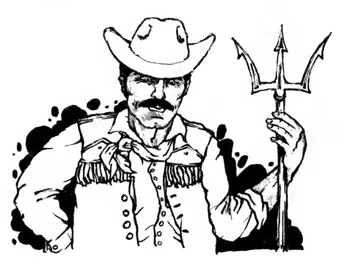
American Power Conversion

По мнению специалистов APC, регион Сибири несомненно представляет больший рынок для продукции фирмы, в первую очередь за счет большего количества пользователей. Продукция APC уже распространяется через дилеров в Кемеровской области.

Конечно, результаты такой масштабной акции не могут сказаться сразу, но по предварительным данным, все участники остались довольны выбором городов, а также достигли определенных результатов. Так, например, компания Merisel заключила договоров на сумму 180 млн. рублей, затратив на организацию тура 20 тыс. долларов. Заключен договор с коммерческой организацией "Кузбасская ярмарка", которая будет заниматься маркетингом для компании Merisel в Кузбасском регионе. Фирма Epson достигла предварительной договоренности о создании сервисной сети в Алма-Ате, а фирма Colorado Memory Systems — заключила дилерские соглашения с рядом коммерческих структур в Алма-Ате. Также, в ходе тура была проведена проверка деятельности уже существовавших дилеров компании Merisel.

В заключение хочется отметить прекрасную организацию работы самого тура (Марина Никитина и Александр Крылов) и выразить благодарность компании Merisel за предоставленную возможность принять участие в бизнес-туре и познакомиться с положением дел на местах.

Компания "Merisel-Компьютерные технологии";
Москва, Крутицкий вал 3, корп. 2,
Тел.: (095) 276-90-08, 276-31-49, 276-31-70,
факс: (095) 274-00-97.
А.Федоров



В ходе выставки Inwecom'92 мне удалось встретиться с президентом фирмы Trident International господином Джоном Хамлетом и услышать от него небольшой рассказ о фирме и ее продукции.

TRIDENT уже пришел

Фирма давно и хорошо известна своими видеоадаптерами, но сфера ее деятельности гораздо шире. Trident занимается разработкой, конструированием и производством современной вычислительной техники и средств телекоммуникаций. В том числе сборкой оборудования фирмы NEC. Компьютеры Trident отличает высокое качество при приемлемой цене. И еще необычный дизайн — эти компьютеры имеют нечасто встречающийся в "сером" мире персоналок черный цвет. Вдобавок ко всему фирма входит в американскую дистрибьюторскую сеть IVACOM.

Игорь Вязаничев: Работает ли фирма в бывших республиках или же только в России?

Джон Хамлет: Мы уже имеем дилеров в большинстве бывших советских республик. У нас есть дилеры в Киеве, Харькове, в Казахстане и, конечно, в России.

И.В.: Сейчас на здешнем рынке трудно с оплатой, как фирма решает эту проблему?

Д.Х.: Во-первых, АО Trident — стопроцентно российская компания. Мы принимаем оплату в рублях, тем самым помогая нашим клиентам в конвертации. Вдобавок мы делаем значительные вложения капитала в России, поскольку большую часть того, что мы зарабатываем, мы оставляем в России.

И.В.: Каким образом фирма тратит заработанные в России рубли?

Д.Х.: Из заработанных денег мы берем достаточно большие суммы, необходимые на оплату помещений, и вообще все платежи в России осуществляются в рублях. Мы также конвертируем рубли и выплачиваем родительской компании. И, конечно, бóльшая часть рублей уходит в инвестиции для развития нашей собственной структуры, чтобы рос наш бизнес здесь, в России.

И.В.: Какие цели вы ставите, когда инвестируете эти деньги?

Д.Х.: Мы с самого начала создавали этот бизнес именно для России. Мы преследовали здесь две цели.

Во-первых, мы собирались производить компьютеры в России и для России, чтобы они изготавливались именно россиянами, по российской технологии, и мы собирались продавать эти компьютеры на Западе тоже.

Во-вторых, мы хотели создать систему дистрибуции на основе дилерской сети. Для обеспечения высокого уровня сервисного обслуживания технологии Trident компания предоставляет дилерам технические консультации. Сейчас мы занимаемся поиском дилеров для этой системы.

И.В.: Да, но организация производства компьютеров требует изрядных капиталовложений в валюте. Как вы решаете эту проблему?

Д.Х.: Я не согласен, почему в валюте? Сейчас мы ищем завод, на котором будем производить значительную часть узлов наших компьютеров. Найдя такой завод, мы бы реконструировали его и установили западное оборудование. И мы готовы инвестировать это. Что касается необходимости валютных капиталовложений, ранее я отмечал, что мы принимаем оплату в рублях, часть которых реинвестируем.

Мы обнаружили, что для новых технологий, которые будут использоваться, для людей, которые будут разрабатывать новую технику, для оборудования для исследований (разработок) не требуется больших дополнительных капиталовложений — все это здесь есть.

Для поддержки технологии, для поддержки людей у нас есть средства. Но это небольшие инвестиции, которые позволят быстро начать выпуск продукции. И, так как мы будем осуществлять основную часть работ в России, много валюты не потребуется.

Компьютерный рынок каждой страны имеет свои особенности. В России мы столкнулись со следующей ситуацией. С одной стороны — высокий технический уровень подготовки персонала. С другой — не очень высокая, по сравнению с западными стандартами, подготовка людей, занимающихся компьютерным бизне-

сом. По всей вероятности, это объясняется тем, что в России пока еще не создана адекватная система подготовки управленческого персонала, которая на Западе создавалась годами.

Второе, что есть в России — огромный рынок. Чудовищный рынок.

На мой взгляд, Россия должна сделать три вещи, чтобы достичь западного уровня в сфере управления бизнесом. Первое — поднять уровень связи до западных стандартов. Второе — улучшить инфраструктуру страны, автоматизировав все те процессы, которые используются в бизнесе. Третье — нужно компьютеризировать и организовать транспортную систему — для всего этого нужны компьютерные технологии.

И.В.: Если можно, поподробнее о ваших планах производства здесь, в России. Производства плат, компьютеров и т.д.

Д.Х.: Во-первых, мы начали переговоры с компаниями, которые разрабатывают и производят корпуса с источниками питания. Во-вторых, несколько компаний выразило готовность производить простейший в технологическом отношении элемент компьютера — видеоадаптер Trident VGA. После этого мы бы хотели начать производство материнских плат — четырех- или шестислойных — и в России есть заводы, которые делают платы такого рода. Мы будем двигаться постепенно — шаг за шагом.

И.В.: Я не совсем понял, на каком заводе это будет производиться: будет ли он построен, или же будет использована чья-то база?

Д.Х.: Как я говорил, мы хотим найти существующий завод и переоснастить его. Мы уже встречались с представителями нескольких электронных заводов, ранее работавших на оборонную промышленность. Сейчас, как вы понимаете, такие заводы практически остановились. У них есть ресурсы, производство, и в большинстве случаев люди, обладающие необходимой профессиональной подготовкой.

И.В.: Уже были разговоры с кем-то конкретно? Смотрели какие-нибудь заводы?

Д.Х.: У нас сейчас начинаются такие переговоры.

И.В.: И каковы первые впечатления?

Д.Х.: У нас есть определенные трудности. В наших отношениях с потенциальными российскими партнерами мы прежде всего ищем полное взаимопонимание, объективность и справедливость. Тот, кто делает наибольший взнос в компанию, берет на себя наибольший риск. Пойти на такой риск можно только в том случае, если вы получаете возможность контроля над управлением компанией. Трайдент пришел в Россию, потому что мы верим в ее будущее. Я не открою вам секрета, если скажу, что мы идем на довольно значительный риск. Именно поэтому нам бы хотелось самим осуществлять управление нашей компанией.

У нас есть мечта, чтобы на компьютерах, продаваемых нами, было написано "Сделано в России" и мы работаем над осуществлением этого.

И.В.: Вы думаете, они ТАМ будут продаваться?

Д.Х.: Да. Если китайцы могут делать компьютеры, и Тайвань, и Гонконг, почему россияне не могут делать компьютеры все лучше и лучше? И я верю, что на одной из выставок Comdex в США мы выставим наши компьютеры под флагом "Сделано в России"!

И.Вязаничев



Об одной ошибке операций с плавающей точкой

НА НЕКОТОРЫХ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ ТИПА IBM PC/AT 286/287 И 386/387, ВЫЯВЛЕННОЙ С ПОМОЩЬЮ MICROSOFT FORTRAN 5.0 И 5.1

На некоторых IBM PC/AT 286/287 и 386/387 обнаружена ошибка выполнения операций с плавающей точкой в программах, корректно разработанных в среде программирования Microsoft FORTRAN 5.0 и 5.1. Диагностические тесты не выявляют ошибок аппаратного обеспечения этих машин. Переход от ускоренного (Turbo) режима эксплуатации к нормальному не приводит к правильному исполнению программ. Сведения об обнаруженной ошибке не содержатся в системной документации [1].

Ошибка возникает в ряде случаев программно-аппаратной обработки незамаскированной исключительной ситуации денормализованного операнда (Denormalized Operand Exception). В результате выполнения загрузочных модулей программ, безошибочно реализованных в исходных модулях и с правильно заданными начальными данными, могут получаться неправильные результаты расчетов или происходить аварийные завершения заданий. Неверные результаты расчетов связаны с закономерным по причине ошибки изменением промежуточных результатов вычислений. Аварийные завершения заданий сопровождаются диагностическим сообщением об ошибке исчерпания стека (Stack UnderFlow). Как следствие, возможны изменение даты в персональном компьютере, потеря информации о типе жесткого диска и затирание CMOS-памяти.

Исследования показывают, что ошибка происходит на стыке системного программного и аппаратного обеспечения и не является ошибкой MS FORTRAN. Любой код, аналогичный коду прикладных программ MS FORTRAN, будет выполняться неправильно. Это установлено в результате тестирования программ, разработанных в среде Microsoft Macro Assembler 5.10, Turbo Pascal 5.5 и TopSpeed MODULA-2 JPI 1.15.

Предложены три способа устранения обнаруженной ошибки на уровне MS FORTRAN и на уровне неправильно исполняющихся загрузочных модулей пользователей.

Характерные примеры ошибки и ее причина

Тестовая программа, детектирующая все характерные последствия обнаруженной ошибки, имеет вид

```
1          Program Bad_Result_Stack_UnderFlow_0_18
2          Real*4 a,d,h
3     1    h=1.0e-40 ! Stack UnderFlow
4          a=6.0
5          d=9.0
6          a=a-h     ! a=0.0
7          d=d+h     ! d=18.0
8          Write(*,*) a, d
9          Go To 1
10         Stop
11         End
```

Ошибка происходит в случае компиляции программы компилятором FL.EXE MS FORTRAN с ключом /FPi87.

В результате выполнения программы будут получаться случайным образом либо правильные значения переменных **a** и **d** (6.0 и 9.0), либо неправильные (0.0 и 18.0) и даже исключительную ситуацию исчерпания стека с аварийным завершением задания. Изменение значений переменных происходит в операторах 6 и 7. Исключительная ситуация порождается оператором 3. Клавиши Ctrl+Break прерывают выполнение программы.

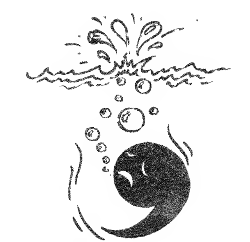
Особенность программы состоит в том, что значение переменной **h** выходит за диапазон представления дей-

ствительных чисел обычной точности (от 1.17E-38 до 3.37E+38). Когда переменная h передается в сопроцессор, возникает исключительная ситуация антипереполнения (UnderFlow Exception). В соответствии с начальной установкой сопроцессора его управляющее слово CW=1332h, и исключительная ситуация замаскирована. Поэтому происходит ее аппаратная обработка способом постепенного (плавного) антипереполнения, в отличие от других ЭВМ, где применяется способ резкого антипереполнения с непосредственным обнулением числа [2, с.123]. Это приводит к исключительной ситуации денормализованного операнда.

Исключительная ситуация антипереполнения в MS FORTRAN замаскирована и может быть размаскирована [3]. Исключительная ситуация денормализованного операнда также замаскирована, но не может быть размаскирована. Создается иллюзия, что эта ситуация разрешается чисто аппаратно — средствами сопроцессора. На самом деле программно-аппаратные действия в персональном компьютере будут другими. Анализ процесса генерации вычислительной среды MS FORTRAN при запуске любой программы показывает, что перед началом выполнения первого исполнимого оператора вместо значения 1332h управляющее слово сопроцессора устанавливается в 1330h. Этот факт скрыт от прикладного программиста. Исключительная ситуация денормализованного операнда становится незамаскированной. Поэтому ее обрабатывает не сопроцессор, а MS FORTRAN своими средствами. Изменение стандартного значения CW=1332h на новое CW=1330h производится следующей группой команд MS FORTRAN:

```
6388:00E6 B83213      MOV   AX,1332              <-- CW=1332h
6388:00E9 E80A05      CALL  05F6
..............................
6388:05F6 A30600      MOV   Word Ptr [0006],AX
6388:05F9 253CFF      AND   AX,FF3C              <-- Маска FF3Ch
6388:05FC A31A00      MOV   Word Ptr [001A],AX
6388:05FF 9B          WAIT
6388:0600 D92E1A00    FLDCW Word Ptr [001A]      <-- CW=1330h
6388:0604 A30A00      MOV   Word Ptr [000A],AX
6388:0607 C3          RET
```

Сброс маски исключительной ситуации денормализованного операнда и обработка прерывания в MS FORTRAN приводит к уточнению мантиссы денормализованного числа.



Функцию обработки незамаскированной исключительной ситуации денормализованного операнда выполняет обработчик MS FORTRAN, который заменяет стандартное аппаратное немаскируемое прерывание INT 02h. При генерации вычислительной среды MS FORTRAN переустановку обработчика прерывания выполняет следующий фрагмент из процедуры __FPINSTALL87:

```
6388:071A BA6207      MOV   DX,0762
6388:071D B80225      MOV   AX,2502
6388:0720 CD21        INT   21
```

В среде отладчика MS CodeView имеем следующий начальный код выполнения тестовой программы для операторов 3, 6 и 7:

```
3:        1   h=1.0e-40 ! Stack UnderFlow
5FD6:0010 CD35067207 INT 35 ;FLD DWord Ptr [___aDBexit+6 (0772)]
5FD6:0015 CD351E7609 INT 35 ;FSTP  DWord Ptr [H (0976)]
5FD6:001A CD3D       INT 3D ;FWAIT

6:             a=a-h     ! a=0.0
5FD6:0034 CD35067609 INT 35 ;FLD   DWord Ptr [H (0976)]
5FD6:0039 CD342E6E09 INT 34 ;FSUBR DWord Ptr [A (096E)]
5FD6:003E CD351E6E09 INT 35 ;FSTP  DWord Ptr [A (096E)]
5FD6:0043 CD3D       INT 3D ;FWAIT

7:             d=d+h     ! d=18.0
5FD6:0045 CD35067609 INT 35 ;FLD   DWord Ptr [H (0976)]
5FD6:004A CD34067209 INT 34 ;FADD  DWord Ptr [D (0972)]
5FD6:004F CD351E7209 INT 35 ;FSTP  DWord Ptr [D (0972)]
5FD6:0054 CD3D       INT 3D ;FWAIT
```

В соответствии с ключом /FPi87 компилятором FL.EXE генерируется код:

```
3:        1   h=1.0e-40 ! Stack UnderFlow
5FD6:0010 90          NOP
5FD6:0011 D9067207    FLD   DWord Ptr [___aDBexit+6 (0772)]
5FD6:0015 90          NOP
5FD6:0016 D91E7609    FSTP  DWord Ptr [H (0976)]
5FD6:001A 90          NOP
5FD6:001B 9B          WAIT

6:             a=a-h     ! a=0.0
5FD6:0034 90          NOP
5FD6:0035 D9067609    FLD   DWord Ptr [H (0976)]
5FD6:0039 90          NOP
5FD6:003A D82E6E09    FSUBR DWord Ptr [A (096E)]
5FD6:003E 90          NOP
5FD6:003F D91E6E09    FSTP  DWord Ptr [A (096E)]
5FD6:0043 90          NOP
5FD6:0044 9B          WAIT

7:             d=d+h     ! d=18.0
5FD6:0045 90          NOP
5FD6:0046 D9067609    FLD   DWord Ptr [H (0976)]
5FD6:004A 90          NOP
5FD6:004B D8067209    FADD  DWord Ptr [D (0972)]
5FD6:004F 90          NOP
5FD6:0050 D91E7209    FSTP  DWord Ptr [D (0972)]
5FD6:0054 90          NOP
5FD6:0055 9B          WAIT
```

Для правильного выполнения программы код отсутствия операции NOP (90h) необходимо заменить на код ожидания процессором выполнения команд сопроцессора WAIT (9Bh) по адресам 5FD6:0015, 5FD6:0039 и 5FD6:004A сразу же за командой FLD. В результате возникновения незамаскированной исключительной ситуации денормализованного операнда в команде FLD (адреса 5FD6:0011, 5FD6:0035 и 5FD6:0046) сопроцессор передает управление в MS FORTRAN на переуста-

новленный обработчик INT 02h. Если замена кода NOP на WAIT не производится, то значение вершины стека регистров сопроцессора может оказаться искаженным перед выполнением прерывания INT 02h:

1) в операторе 3 вершина стека пуста, а не h;
2) в операторе 6 вершина стека содержит значение a=6.0, а не h;
3) в операторе 7 вершина стека содержит значение d=9.0, а не h.

Причина искажения состоит в том, что в сопроцессоре может не фиксироваться возникновение исключительной ситуации денормализованного операнда в команде FLD. Вместо этого происходит выполнение следующей команды сопроцессора. Только здесь фиксируется исключительная ситуация, и для нее предпринимаются ответные действия в переустановленном обработчике INT 02h. Обработчик INT 02h дешифрует команду, в которой возникла исключительная ситуация, и выполняет ее с откорректированными операндами. Поэтому повторное выполнение неправильно зафиксированной команды в теле обработчика INT 02h приводит к тому, что:

1) в операторе 3 команда FSTP INT 02h вызывает исключительную ситуацию исчерпания стека с аварийным завершением задания;
2) в операторе 6 команда FSUBR INT 02h вычисляет a=0.0;
3) в операторе 7 команда FADD INT 02h вычисляет d=18.0.

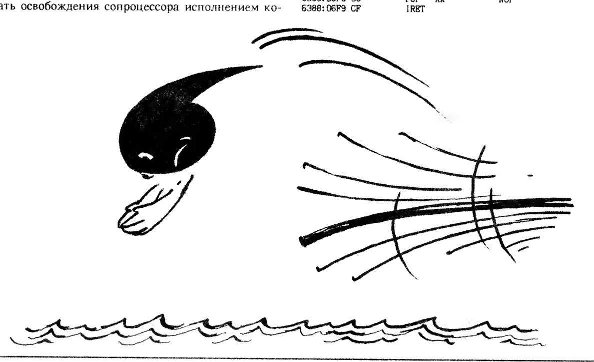
Анализ взаимодействия процессор-сопроцессор и характер ошибочного выполнения тестовой программы позволяют предложить три способа устранения ошибки средствами MS FORTRAN:

- непосредственное указание процессору каждый раз ожидать освобождения сопроцессора исполнением команд WAIT вместо команд NOP, разделяющих соседние команды сопроцессора. Способ является наиболее корректным. Он не затрагивает идеологии создания загрузочных модулей в MS FORTRAN — управляющее слово сопроцессора вместо 1332h по-прежнему устанавливается в 1330h, и стандартный обработчик прерывания INT 02h переназначается на внутренний обработчик MS FORTRAN;
- задание нового управляющего слова сопроцессора CW=1333h, которое маскирует биты исключительных ситуаций денормализованного операнда и недействительной операции. В результате эти исключительные ситуации будут обрабатываться чисто аппаратно — средствами сопроцессора без задействования переустановленного обработчика INT 02h;
- запрещение переустановки прерывания INT 02h на внутренний обработчик MS FORTRAN. Так как по-прежнему CW=1330h, все незамаскированные исключительные ситуации операций с плавающей точкой будут обрабатываться стандартным обработчиком INT 02h MS-DOS.

Устранение ошибки на уровне MS FORTRAN и загрузочных модулей пользователей

При генерации в MS FORTRAN кода прикладных программ со встроенными инструкциями сопроцессора используется одинаковый фрагмент кода из библиотек LLIBFOR7.LIB и LLIBFORE.LIB:

```
6388:06F1 C60490    MOV   Byte Ptr [SI],90
6388:06F4 5B        POP   BX        ↑
6388:06F5 5E        POP   SI        |
6388:06F6 1F        POP   DS     генерируется
6388:06F7 5D        POP   BP   команда ассемблера
6388:06F8 58        POP   AX         NOP
6388:06F9 CF        IRET
```



Чтобы исправить на уровне MS FORTRAN ошибку программно-аппаратной обработки денормализованных чисел, необходимо с помощью утилиты редактирования двоичных файлов (например, из пакета PCTools) в библиотеках LLIBFOR7.LIB и LLIBFORE.LIB фрагмент кода

```
C60490
```

заменить на

```
C6049B
```

В результате замены все операции с сопроцессором будут генерироваться с использованием кода

```
6388:06F1 C6049B      MOV   Byte Ptr [SI],9B
6388:06F4 5B          POP   BX          ↑
6388:06F5 5E          POP   SI          |
6388:06F6 1F          POP   DS       генерируется
6388:06F7 5D          POP   BP    команда ассемблера
6388:06F8 58          POP   AX           WAIT
6388:06F9 CF          IRET
```

и каждой команде сопроцессора будет предшествовать команда WAIT.

В "сбойных" загрузочных модулях пользователей выполняется аналогичная замена. Для сохранения правильной контрольной суммы модулей из значения 19-го байта их заголовков вычитают 0Bh.

При реализации второго способа устранения ошибки в библиотеках MS FORTRAN фрагмент кода

```
253CFF
```

заменяется на

```
0D0300
```

В результате вместо команды

```
6388:05F9 253CFF      AND   AX,FF3C
```

будет исполнена команда

```
6388:05F9 0D0300      OR    AX,0003
```

Управляющее слово сопроцессора CW примет значение 1333h. Исключительные ситуации денормализованного операнда и недействительной операции оказываются замаскированными и будут обрабатываться аппаратно средствами сопроцессора, а не программно средствами MS FORTRAN.

В третьем способе в библиотеках MS FORTRAN фрагмент кода

```
B80225CD21
```

заменяется на

```
B802259090
```

Поэтому вместо команд

```
6388:071D B80225      MOV   AX,2502
6388:0720 CD21        INT   21
```

будут исполнены команды

```
6388:071D B80225      MOV   AX,2502
6388:0720 90          NOP
6388:0721 90          NOP
```

Переустановки обработчика прерывания INT 02h не произойдет. Так как управляющее слово сопроцессора осталось прежним (CW=1330h), обработка незамаскированных исключительных ситуаций операций с плавающей точкой будет производиться стандартным обработчиком INT 02h.

Альтернативой приведенным способам является использование на этапе компиляции программ ключей FL.EXE, отличных от /FPi87.

Ошибочного выполнения приведенной тестовой программы не будет происходить, если перед ее исполнением замаскировать 5-й бит регистра масок прерываний второго контроллера прерываний 8237А-5 через порт 0A1h. В результате не произойдет вызов прерывания INT 75h, которое в схемах IBM PC/AT вызывает стандартное прерывание INT 02h. Функцию маскирования выполняет следующий фрагмент:

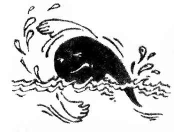
```
MOV   DX,00A1
IN    AL,DX
OR    AL,20
MOV   DX,00A1
OUT   DX,AL
```

Указанным приемом не рекомендуется пользоваться ввиду полного запрещения обработки прерываний от математического сопроцессора.

Тестирование других персональных компьютеров

"Ошибочный" загрузочный модуль тестовой программы и модули, "исправленные" в соответствии с приведенными рекомендациями, были исполнены на семи IBM PC/AT 286/287. Ошибочное выполнение программы выявлено на четырех из них. "Исправленные" варианты, соответствующие каждому из предлагаемых способов, функционируют на этих машинах без сбоев. "Ошибочный" и "исправленные" модули отработали правильно на остальных трех машинах. Тот же эффект наблюдался и на 386/387 машинах, однако у авторов не было возможности набрать здесь достаточной статистики.

Выводы



Обнаруженная ошибка не является ошибкой транслятора или библиотек MS FORTRAN. Критические участки кода "ошибочной" программы свидетельствуют, что транслятор и редактор связей сгенерировали теоретически правильный код, который не должен приводить к некорректностям операций с плавающей точкой. Ошибка возникает на уровне микропрограммного управления и связана со срывом в работе процессор-сопроцессор для ряда персональных компьютеров. Тестирование их процессоров и математических сопроцессоров известными диагностическими тестами не выявляет эксплуатационных программно-аппаратных сбоев. Од-

нако такие машины могут неправильно выполнять операции с денормализованными числами. Ошибка состоит в том, что если в первой из двух последовательных команд сопроцессора возникает исключительная ситуация денормализованного операнда, то она может не фиксироваться. Происходит выполнение второй команды сопроцессора, и только здесь прекращается выполнение программы и обнаруживается исключительная ситуация. Поэтому, если в прерывании, обрабатывающем исключительную ситуацию, предусмотрена корректировка операндов команды и ее повторное выполнение, это приводит к двойному выполнению команды и неправильным результатам. Для правильного выполнения программы между командами сопроцессора необходимо поместить команду WAIT.

Наличие обнаруженной ошибки на четырех компьютерах различных фирм-изготовителей указывает на малую вероятность ее возникновения в результате технологических ошибок сборки. Вероятно, для выяснения истинных источников ошибки необходимо более детальное исследование фирмами-разработчиками архитектуры и взаимодействия процессор-сопроцессор.

А.Дячкин, В.Парамонов
Тел.: (095) 263-63-10, 528-54-36

Литература:

1. 80386 B-1 Stepping. Specification Changes and Errata. 80387 B-0 Stepping. Documentation Updates, Specification Changes and Errata. — INTEL Technologies Inc., 25 p.
2. Морс С.П., Алберт Д.Д. Архитектура микропроцессора 80286. — М.: Радио и связь, 1990. — 300 с.
3. Microsoft FORTRAN. Advanced Topics. Version 5.0. For MS OS/2 and MS-DOS Operating Systems. — Microsoft Corporation, 301 p.



Инструментальные возможности открытой текстовой среды

Разработка современных прикладных систем начинается и заканчивается интерфейсами. Вообще-то это парадокс: при выполнении сложных и глубоких математических вычислительных алгоритмов пользователь довольствуется числовыми выдачами, а у простейшей по существу задачи ввода и редактирования текста интерфейс требует полсотни пунктов меню, пару десятков диалоговых боксов, довольно сложную панель управления. Впрочем, все здесь ясно: круг пользователей математической задачи — несколько ученых чудаков, а за клавиатурой текстового процессора сидят многочисленные пользователи самых разных квалификаций.

Для программиста — разработчика прикладных систем это означает, что основной объем трудозатрат перетекает из области информационно-расчетного содержания системы в область ее интерфейсной формы. Ясно, что такая тенденция должна как-то отразиться и на инструментальных средствах, применяемых для разработок. Попробуем сформулировать основные принципы проявления этих тенденций. Они, на наш взгляд, таковы:

- визуальное, средоориентированное программирование;
- объектный характер среды;
- использование встроенного языка программирования;
- открытость архитектуры.

Рассмотрим несколько подробнее каждый из этих принципов.

Визуальное программирование

Визуальное программирование — это довольно новая технология разработки, которая отвечает доминирующей роли интерфейсов в архитектуре прикладных систем. Визуальное программирование — это программирование, осуществляемое прямо в той среде, которая является целью разработки. Работа визуального программиста отличается от работы процедурного программиста так же, как работа мастера на строительном участке отличается от работы архитектора, проектирующего строительство. Визуальный программист не описывает на каком-либо языке, как должна выглядеть и что должна делать система (как описывается здание на чертежах архитектора), а подобно мастеру находится прямо на строительном участке, на котором возводится здание.

Пользуясь этой метафорой, определим основные компоненты визуального программирования:

- *строительная площадка* — это изначально пустая визуальная среда, в которой могут размещаться и функционировать разные объекты — элементы прикладной системы;
- *строительные блоки* — классы визуальных объектов, из экземпляров которых — кнопок, пунктов меню, диалоговых боксов, текстовых полей и прочих — строятся визуальные среды прикладных систем;
- *строительный раствор* — встроенный язык программирования, служащий средством для связывания объектов друг с другом, придания им нужной формы путем создания специализированных подклассов визуальных объектов, для описания их внешнего вида, диалогового поведения;
- *строительные механизмы* — инструменты, служащие для порождения, передвижения, изменения, присоединения друг к другу визуальных объектов.

Объектный характер среды

Процесс такого визуального программирования возможен лишь при условии, что среда не описывается процедурно, а составляется из независимых друг от друга визуальных объектов — кнопок, меню, текстовых полей и т.п. Каждый такой объект несет в себе какую-то содержательную информацию и сам заботится о правильности своего внешнего вида, о соответствии этого внешнего вида внутреннему состоянию, о реагировании на сообщения, получаемые от других объектов

и внешнего окружения — с клавиатуры, от мыши, таймера, коммуникационных каналов.

Процедуры, обеспечивающие активность системы, не объединяются в монолитные супералгоритмы (как это было в прежней необъектной технологии), а присоединяются к тем или иным объектам (точнее, к их классам).

Только это и делает возможным непосредственное визуальное конструирование: достаточно выбрать объект и поставить его в определенное место среды — и он уже может существовать и действовать в этой среде без переписывания и отладки какого-то общего алгоритма.



Встроенный язык программирования

Важнейшую роль в визуальной технологии играет встроенный язык программирования. Без него визуальный конструктор был бы немногим больше, чем "рисовалка", в которой изображением были бы не простые картинки, а многокнопочные экраны. Такими кнопочными рисовальщиками являются редакторы интерфейсных объектов типа Whitewater Resource Toolkit. Они очень неплохо справляются с процессами создания и расстановки визуальных объектов, но не в состоянии вдохнуть в эти объекты жизнь. Создаваемые с помощью этих инструментов визуальные среды остаются статичными до тех пор, пока на языке программирования С не будет написана программа, принимающая сигналы от кнопок и реагирующая на них.

Некоторые визуальные системы типа ObjectVision или ToolBook подходят к решению этого вопроса путем создания все более высокоуровневых классов визуальных объектов, наглядно представляющих те или иные связи в среде (гиперссылки). Это конечно же правильный и перспективный путь. Чем больше удачных понятий и метафор такого рода будет найдено, тем в большей степени мы сможем избавить пользователя от программирования.

Однако сколь богатые метафоры ни привлекай к изображению информационной среды и связей внутри нее, сколь выразительные классы визуальных объектов ни давай пользователю, всегда останутся какие-то связи, зависимости, реакции, требующие не образно-визуального, а алгоритмического задания.

В то же время действия очень многих кнопок и пунктов меню описываются столь тривиальными формулами и операторами, что переключаться ради них в полноценную среду процедурного программирования как-то очень не хочется. Двадцатисимвольная строчка, описывающая действие кнопки, при этом отрывалась бы от самой кнопки, теряясь где-то в алгоритмических глубинах прикладной системы.

Намного лучше и удобнее, когда у всякой кнопки можно приоткрыть потаенную дверцу и на вывесившийся из нее лист нанести текст короткой формулы, а то и большой программы, после чего кнопка немедленно обретает поведение, описываемое введенной формулой. Ее можно тут же и нажать, проверяя и отлаживая работоспособность.



Открытость архитектуры

Открытость — это наиболее общее современное технологическое требование. В отношении систем визуального конструирования суть его состоит в том, чтобы позволить вносить в конструируемую среду не только объекты из какого-то фиксированного набора, но и внешние объекты. Так, если некто сконструировал хороший визуальный калькулятор, то другой должен иметь возможность легко взять его, как с полки, и внести в свою среду в качестве внутреннего элемента или использовать рядом со своими собственными подобными диалоговыми объектами.

State of Art

Сегодня идеология и технология визуального программирования одновременно и богата и бедна.



Крупнейшая информационная сеть СНГ даст Вам возможность общения с любым абонентом всемирных сетей Internet, Bitnet, MCI-Mail, Compuserv, EUnet и многих других.

Ваше сообщение, программа или графический файл дойдет до любой точки мира меньше чем за 4 часа! В отличие от факса, не будет потеряно ни одной буквы.

Электронная почта — современнейшая технология общения. Теперь она доступна каждому.

Используя возможности сети RELCOM, Вы можете обсудить проблему со всеми заинтересованными в ее решении людьми по всей планете одновременно, Вы можете извлечь информацию из сотен банков данных, получить свежую версию программы, даже рекламировать свою продукцию и услуги.

В сети RELCOM активно работают многие биржи, банки, крупные и небольшие предприятия, институты — всего более 60,000 на территории СНГ и далеко за миллион во всем мире. У абонентов RELCOM появляется возможность прямой связи с большинством крупных компьютерных, программистских и электронных фирм.

Единственный недостаток RELCOM — однажды начав работать в ней, Вы будете делать это всю жизнь.



(095) 231-21-29, 231-63-95, 233-06-70

Ее богатство проявляется в уже весьма большом количестве реальных систем визуального программирования. Это и Visual Basic, и Actor, и ObjectVision, и Knowledge Pro и ToolBook и некоторые другие. К этому же ряду относится и *Инструментальный Мастер* с его технологией средоориентированного (читай — визуального) программирования.

И несмотря на достаточное обилие этих и других систем, визуальное программирование все еще не обрело такого же гармоничного и концептуально сбалансированного выражения, какое обрело, скажем, процедурное программирование в языках Pascal и С, функциональное программирование в языке Lisp, логическое программирование в языке Prolog, объектное программирование в языке SmallTalk. Поэтому продолжаются поиски более гармоничной концептуальной платформы визуального программирования, и появляются новые инструментальные системы, поддерживающие его в самых разных формах.

В проекте разработки *Визуального Мастера* мы тоже делаем такую попытку. Этот проект обобщает опыт средоориентированного программирования, накопленный за несколько лет разработки и использования *Инструментального Мастера*. В то же время помимо собственно концептуальных решений в этом проекте преодолеваются многие чисто технические ограничения, присутствие которых в *Инструментальном Мастере* было обусловлено тогдашним уровнем возможностей.

Что такое Визуальный Мастер?

С точки зрения потребителя *Визуальный Мастер* (ВМ) — это интерактивная среда, в которой могут существовать визуальные объекты разнообразных классов и действует мощный встроенный язык программирования, называемый *Мастер*.

Вместе с этой средой пользователю предлагается набор средств, представленных в виде тех или иных визуальных объектов, из которых одни предназначены для обслуживания конечных прикладных задач (таких как редактирование текстов или работа с электронными таблицами и с базами данных), а другие имеют инструментальный характер и служат для поддержки процессов создания новых прикладных объектов и сред.



С концептуальной же точки зрения в системе ВМ необходимо выделить следующие аспекты, как наиболее принципиальные:
- абстрактность визуальной среды;
- исчерпывающее объектное представление;
- глубокий симбиоз визуальной среды и языка программирования.

Охарактеризуем каждый из этих аспектов поподробнее.

Абстрактность визуальной среды

С точки зрения внутреннего устройства визуальная среда в ВМ имеет тот же характер, что и любая другая современная визуальная среда подобная Windows. Это значит, что весь интерфейс строится из объектов различных классов: кнопок, диалоговых боксов, группирующих рамок, окон с содержательной информацией и управляющими панелями. Из этих объектов составляется информационное содержание любой системы. Операционная организация системы основана на понятии сообщения. Вместо управляющей программы инициатива отдана приходящему извне потоку сообщений о событиях на клавиатуре, мыши, таймере и т.п. На такие сообщения реагируют различные объекты, руководствуясь логикой, запрограммированной в соответствующем классе объектов.

В системе ВМ визуальная среда организована именно таким объектным образом, и точно так же она управляется не внутренней программой, а внешним потоком событий. Классы объектов определены таким образом, чтобы не зависеть ни от какой конкретной операционной платформы — DOS, Windows, OS/2 Presentation Manager и др.

Предполагается постепенно распространить ВМ на разные системные платформы и сделать его объекты доступными и переносимыми между MS-DOS, Windows, OS/2 Presentation Manager и др. Начиналась же разработка с системной платформы MS-DOS, где интерактивная среда должна была создаваться "с нуля". Необходимо было создать и монитор виртуальной памяти, и менеджер классов, обеспечить динамическую

загрузку DLL-модулей, управление окнами — визуальными объектами, разработать набор классов элементов пользовательского интерфейса — кнопок, окон, меню и проч. В Windows и других оконных платформах используется, естественно, стандартная интерактивная среда, расширяемая лишь более высокоуровневой объектной поддержкой и встроенным языком программирования.

Исчерпывающее объектное представление

Главным концептуальным решением в ВМ является последовательный объектный подход к формированию всех элементов прикладной системы. На наш взгляд недостаточно, чтобы кнопки, меню, диалоговые боксы и прикладные окна всего лишь обрабатывались как объекты во время работы системы. Необходимо, чтобы их можно было записать на диск в объектной форме, которая позволит и передать их в другой сеанс работы, и вставить в качестве компонентов в другую визуальную среду, и перенести на другую системную платформу (например, из MS-DOS в Windows). Эта проблема известна в информационных объектно-ориентированных системах как проблема построения persistent objects.

Отчасти подобная форма представления визуальных объектов реализована в качестве стандартного средства в Windows — файлов ресурсов. В них, однако, можно записывать лишь статическую информацию — координаты, размеры, цвета, текстовые надписи и другие числовые и символьные атрибуты визуальных объектов. Самое же главное — это класс объекта с присоединенными к нему методами доступа, который при стандартном подходе останется существовать не в объектной, а в процедурной форме — в программном коде, рассредоточенном по EXE-модулю прикладной системы и неотторжимом от нее.

Обычно исчерпывающее объектное представление возможно лишь для объектов, встроенных в систему классов, — кнопок, меню, диалоговых боксов, — и не распространяется на прикладные классы объектов, которые как раз и представляют собой цель работы проектировщика прикладной системы. В ВМ исчерпывающий характер объектного представления основывается на том, что сами классы объектов и присоединенные к ним методы доступа тоже представляются в виде объектов; они могут сохраняться среди и внутри обычных прикладных объектов.

Представимость классов объектами достигается за счет использования специального пакета объектной поддержки — менеджера объектов. Функции и методы доступа представимы в виде объектов, независимых ни от конфигурации системы, ни от типа системной платформы, потому что они запрограммированы на встроенном языке программирования.

Мало того, что классы, методы и функции представлены в виде объектов. В системе ВМ эти объекты-классы, объекты-методы и объекты-формулы являются такими же *визуальными* объектами, как и кнопки, окна, диалоговые боксы. Конечному пользователю это, разумеется, не нужно и не видно, а разработчику прикладной системы это очень важно, поскольку позволяет создавать, изменять и уничтожать их прямо в той среде, которую он разрабатывает, что предельно сокращает цикл проектирования и отладки.

Встроенный язык Мастер

Второе принципиальное отличие ВМ — это особый по характеру встроенный язык программирования. С одной стороны, по синтаксической форме этот язык сделан максимально простым и традиционным. Он не требует от программиста привыкания к новым синтаксическим формам как, скажем, в Lisp, Prolog, APL или SmallTalk. Синтаксис и процедурная семантика языка вполне Pascal- или С-подобная.

С другой стороны, необычность языка заключается в способе его интеграции с охватывающей визуальной средой. Среда составляет с языком единое целое из-за того, что она служит для него и глобальным информационным контекстом, и структурообразующим средством.

В обычных языках программирования подобная проблема решается синтаксически: тексты функций вложены в тексты модулей, где в текстовом же виде определены все типы данных и глобальные объекты, используемые функциями, а сами модули образуют макроструктуру программной системы.

В ВМ текстовое представление заканчивается на отдельных функциях и формулах. Эти тексты в виде объектов разбросаны по визуальной среде, вставлены на разной глубине в другие объекты, и эта расстановка как раз и задает ту общую структуру программной системы, которая при обычном программировании представлялась бы текстом компилируемого модуля и MAKE-файла.

Все объекты в среде (кнопки, меню, окна, тексты, диалоговые боксы) имеют внутренние имена, и функции "видят" эти имена в соответствии с естественным контекстным правилом: если имя, такое как MinButton, Client или VerScrollBar, встречается в нескольких местах, то функция при упоминании такого имени связывает его с пространственно ближайшим объектом. Таким образом, расположение объекта-функции в среде существенно определяет его семантику.

Поскольку функции представляются как равноправные объекты, то при записи охватывающей среды на диск в виде объекта они записываются в составе этого объекта, что и делает его исчерпывающим, так как помимо статических данных в нем оказывается и функциональная информация.

Впрочем это не означает, что все программирование в системе обязано вестись только на встроенном языке *Мастер*. Наоборот, предполагается довольно активное использование и низкоуровневых языков С или Pascal. На этих языках можно запрограммировать какие-то ал-

горитмы, требующие сложной структурной обработки информации или высокой вычислительной эффективности. Функции и классы, реализованные таким образом, легко доступны из языка *Мастер* и из визуальной среды через механизмы динамической загрузки DLL-модулей. А с концептуальной точки зрения и эти функции, и классы тоже представлены объектно, но уже не полными телами, а небольшими объектами-дескрипторами, ссылающимися на соответствующий DLL-модуль.

Другие элементы технологии

Понятно, что сказанное характеризует лишь основное концептуальное ядро системы. Технологически в ней имеется множество других компонентов. Прежде всего следует отметить монитор виртуальной памяти VMM, который позволяет на платформе MS-DOS (без Windows) использовать XMS/EMS-расширения памяти и защищенный режим 286/386 процессоров, а также файловые виртуальные расширения памяти. Этот же монитор выполняет и поддержку программирования в формате DLL (Dynamic Link Libraries — динамически компонуемых библиотек), что делает всю архитектуру совершенно открытой и практически совмещаемой с Windows.

В системе имеются мощные пакеты для обработки текстов, графики, электронных таблиц, для абстрагированного доступа к базам данных (в частности, в форматах BTRIEVE, dBASE).

Коммерческое представление продуктов

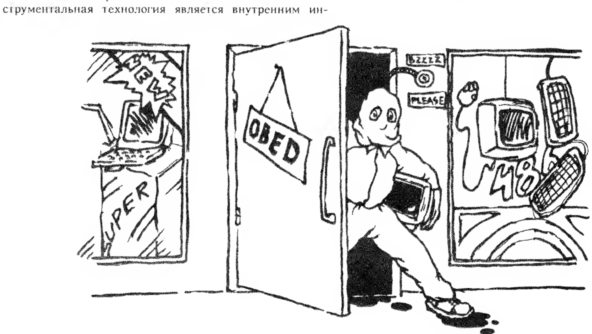
В настоящее время ВМ и вся связанная с ним инструментальная технология является внутренним инструментом группы разработчиков, работающих в фирме МИКРОИНФОРМ. В виде самостоятельного продукта она будет представлена только после завершения таких технологических модулей, как огладчик, визуальный конструктор, и других вспомогательных инструментальных средств, а главное — после создания исчерпывающего комплекта документации.

Пока же с помощью этой технологии фирма МИКРОИНФОРМ разрабатывает собственные прикладные системы, среди которых основной является новая версия текстового процессора *Лексикон-2*. В *Лексикон-2* мы стремимся перенести от его предшественника *Лексикона-1* простоту и привычность текстовой модели и добавляем давно ставшие необходимыми стили, графику, шрифты, ряд вспомогательных утилит.

Но более важно то, что в новом воплощении текстовый процессор *Лексикон-2* оказывается не замкнутой системой, служащей лишь для текстовой обработки, а открытой средой, к которой легко подключаются как небольшие полезные подспорья (калькуляторы, электронные часы и календари, файлеры и т.п.), так и более серьезные расширения, позволяющие работать с таблицами, получать доступ к базам данных, выполнять какие-то специализированные прикладные действия, вплоть до уровня экспертных систем.

В сущности, сам *Лексикон-2* — это всего лишь один из конкретных диалоговых объектов-сред в открытой визуальной инструментальной среде ВМ. И если одним конечным пользователям будет достаточно именно этих конкретных текстовых возможностей, то другие могут воспользоваться *Лексиконом-2* как инструментальной платформой для создания собственных прикладных систем. В этом органичном сочетании прикладного и инструментального направлений состоит сегодняшняя стратегия разработки МИКРОИНФОРМ.

Е.Веселов





Локальная сеть и шахматы, или Почему вы не используете NetBIOS?

Использование NetBIOS

Эмуляторы NetBIOS

Как отмечалось, эмуляторами NetBIOS снабжены практически все сетевые ОС ведущих фирм. Мы ограничимся рассмотрением использования эмуляторов NetBIOS в двух системах — LANsmart и NetWare.

Эмулятор NetBIOS лежит в основе сетевой ОС LANsmart фирм D-Link и Datex Inc., поэтому для его применения не требуется никаких дополнительных шагов.

В NetWare для использования NetBIOS необходимо сначала запустить на рабочей станции интерфейсный модуль IPX.COM, а затем и сам эмулятор NETBIOS.EXE. После этого прикладные программы, использующие NetBIOS, могут нормально выполняться.

Пример применения NetBIOS — утилита "шахматный запараллеливатель"

Настала пора предъявить, наконец, читателю обещанный в начале статьи пример применения NetBIOS в прикладной программе. Речь пойдет, как уже было сказано, об утилите Chess Paralleler (CHPRL), позволяющей пользователям локальной сети играть в шахматы, применяя популярную программу Chess Master. Как известно, Chess Master имеет режим работы, при котором два игрока могут сыграть в шахматы друг с другом, используя компьютер как интеллектуальную шахматную доску. Играть вдвоем за одной клавиатурой и одним монитором не всегда удобно. CHPRL логически " запараллеливает" клавиатуры двух рабочих станций локальной сети путем занесения в буфер клавиатуры удаленного компьютера скан-кода и соответствующего ASCII-кода (или расширенного ASCII-кода и нуля), введенного с клавиатуры локального компьютера. Поясним сказанное следующим примером. Пусть есть две рабочие станции, на которых резидентно установлена CHPRL, обозначим их А и В. Предположим, что пользователь нажал некую клавишу на клавиатуре локальной (для него) станции А. CHPRL, контролирующая ряд векторов, и в частности вектор 9 (вектор прерываний от клавиатуры), перехватывает скан-код и ASCII-код, сгенерированные по нажатию на клавишу (не препятствуя, однако, их нормальному попаданию в буфер клавиатуры станции А) и, используя NetBIOS, пересылает их своему партнеру по обмену данными — утилите CHPRL на удаленной рабочей станции, которая в свою очередь занесет их в буфер клавиатуры своей рабочей станции (станции В). В результате этих действий эмулируется нажатие той же клавиши на клавиатуре рабочей станции В. Аналогично при нажатии другим пользователем некой клавиши на клавиатуре локальной (для него) станции В скан-код и ASCII-код, соответствующие этой клавише, окажутся также в буфере клавиатуры удаленной станции А, благодаря чему эмулируется нажатие той же клавиши на ее клавиатуре. Отсюда видно, что принцип действия CHPRL основан на принятом в ROM BIOS механизме буферизации вводимых с клавиатуры данных с применением кольцевого буфера. Для читателей, не знакомых с этим механизмом, мы подробнее остановимся на нем несколько позже. А теперь рассмотрим, как используется интерфейс NetBIOS для обмена рабочих станций скан-кодами и ASCII-кодами.

Для обмена данными между рабочими станциями А и В на каждой из них (в теле резидентной утилиты CHPRL) предусматривается по два блока NCB, один из них (NCBI) предназначен для приема данных, другой

Окончание. Начало в КомпьютерПресс №№ 2,3'93.

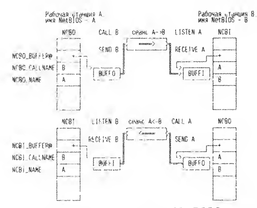


Рис. 1. Использование NetBIOS для обмена данными между утилитами CHPRL на рабочих станциях A и B

(NCBO) — для их передачи (рис. 1). С блоками NCB связываются два буфера длиной в слово — BUFFI для приема и DUFFO для передачи. Рабочим станциям ставятся в соответствие имена NetBIOS, например те же A и B. Между этими именами открываются два сеанса: для передачи данных от A к B (сеанс A-»B) и для передачи данных от B к A (сеанс A«-B). Для открытия сеанса A-»B имя A выдает команду CALL для имени B, а имя B, в свою очередь, — команду LISTEN для имени A. Для открытия сеанса A«-B все происходит в обратном порядке.

Процедура открытия сеансов A-»B и A«-B обладает некоторыми особенностями, обусловленными противоречием между симметрией взаимодействующих процессов (взаимодействуют две одинаковые утилиты CHPRL) и асимметрией протокола открытия сеанса (взаимодействующие процессы должны выдавать разные команды: с одной стороны — LISTEN, с другой — CALL, причем LISTEN должна быть выдана раньше), а также отсутствием какой-либо синхронизации по времени при запуске утилит CHPRL. На рис. 2 приведена упрощенная блок-схема простого и достаточно эффективного алгоритма процедуры установления сеансов, примененного в утилите CHPRL. Для наглядности в блок-схеме подставлены конкретные имена для работы на станции A: свое имя — A, чужое — B. Применительно к рабочей станции B имена в блок-схеме следует поменять местами.

Как видно из блок-схемы, после определения своего имени (A) командой ADD NAME выдаются команды LISTEN и CALL для имени-партнера (B) в режиме no-wait (поле NCB_POST@ установлено в 0). Затем в цикле из 50 повторений проверяется, завершились ли выданные команды с кодом возврата 0 (проверяется поле NCB_CMD_CPLT). Если да, то требуемые сеансы A-»B и A«-B установлены и следует выйти из цикла. В противном случае по очереди проверяется, завершились ли команды LISTEN и CALL, и если соответствующая команда завершилась, то проверяется результат ее выполнения. Если команда еще не завершилась (поле NCB_CMD_CPLT=FFH), то следует подождать ее завершения, для чего организуется задержка, перед которой на экран выводится точка в качестве признака отсутствия связи между утилитами CHPRL. Если команда завершилась, но код возврата не 0 (что скорее всего вызвано отсутствием в сети имени-партнера или, в случае команды CALL, тем, что оно еще не выдало команду LISTEN), команду следует выдать повторно. На первый взгляд может показаться, что проверка кода возврата на 0 в этом месте не нужна, однако это не так. Дело в том, что команда, еще не завершившаяся при проверке кода возврата на 0 выше по блок-схеме, к тому моменту могла завершиться. Следовало бы, вообще говоря, проанализировать ненулевой код на предмет неисправимого сбоя сетевого адаптера, чтобы избежать бесполезных повторений, что действительно делается в утилите (как и некоторые другие, опущенные на блок-схеме, проверки).

Каждый раз при обнаружении отсутствия связи очередная точка выводится на экран правее предыдущей, в результате на экране образуется удлиняющаяся строка точек. Если после вывода строки из 50 точек тре-

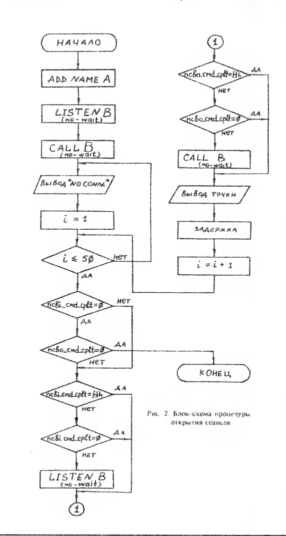
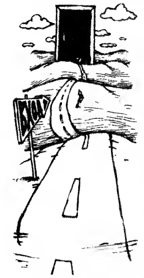


Рис. 2. Блок-схема процедуры открытия сеансов

буемые сеансы по-прежнему не открыты, описанные действия повторяются. Заметим, что в ту же строку, куда выводятся точки, каждый раз выводится фраза "NO CONNECTION" (связь не установлена), предварительно стирающая строку. После того как сеансы установятся, эта строка затирается последующими сообщениями. Описанный выше способ динамической индикации попыток установления связи, разумеется, не претендует на оригинальность и приведен здесь в качестве примера возможного поведения использующей NetBIOS программы при открытии сеанса.



Прежде чем перейти к рассмотрению основного алгоритма CHPRL, поговорим о буфере клавиатуры и его использовании клавиатурными процедурами ROM BIOS. Таких процедур две. Одна из них обрабатывает прерывания от клавиатуры, возникающие после нажатия и отпускания любой клавиши, другая — прерывания, вызываемые командой INT 16H, выданной из программ, использующих клавиатурный сервис ROM BIOS.

Процедура обработки прерываний от клавиатуры считывает скан-коды из соответствующего регистра клавиатуры, распознает нажатие таких служебных клавиш, как CapsLock, NumLock, ScrollLock, и клавиш в сочетании, например, Shift-A, Ctrl-C или Alt-X, генерирует требуемые ASCII-коды и расширенные ASCII-коды для сочетаний типа Shift-F1 или Ctrl-ENTER. Полученные в результате скан-коды и ASCII-коды помещаются в буфер клавиатуры. Последний представляет собой область, содержащую 16 слов, и находится в сегменте данных ROM-BIOS по адресу 40H:1EH.

Если процедура обработки прерываний от клавиатуры является "заполнителем" буфера клавиатуры, то процедура обработки прерываний по INT 16H — это напротив, "извлекатель" данных из буфера[2]. Для обеспечения независимости процессов занесения и извлечения данных из буфера в ROM BIOS используется специальный механизм буферизации, основанный на применении двух указателей длиной в слово. Один из них, расположенный по адресу 40H:1AH, называется указателем на голову буфера, другой, расположенный по адресу 40H:1CH,— указателем на его хвост. Для краткости их обычно называют просто головой и хвостом. Голова буфера всегда содержит смещение слова в буфере (скан-код в старшем байте этого слова, ASCII-код — в младшем), которое будет извлечено по следующему INT 16H, а хвост буфера — смещение слова, в которое будут занесены скан-код и ASCII-код, полученные в результате очередного прерывания от клавиатуры. По мере занесения данных в буфер хвост продвигается в сторону возрастания адресов, в эту же сторону по мере извлечения из буфера данных продвигается и голова. Шаг продвижения указателей равен 2. В процессе занесения и извлечения данных голова буфера обычно "гонится" за хвостом до тех пор, пока не настигнет его, что происходит при равенстве указателей и означает, что буфер пуст. Когда в процессе продвижения голова или хвост оказываются за физическим концом буфера, они вновь устанавливаются на начало, благодаря чему буфер логически "сворачивается" в кольцо, из-за чего его часто называют кольцевым.

Особая ситуация возникает, если данные заносятся, но не извлекаются из буфера. В этом случае в результате продвижения хвост "заворачивается", становясь меньше головы и, в пределе, оказывается перед ней (на слово ближе к началу буфера). Возможен также вариант, когда хвост указывает на конец буфера, а голова — на начало. И в том, и в другом случае буфер оказывается полностью заполненным, и на любую попытку ввести новые данные с клавиатуры ROM BIOS реагирует звуковым сигналом, а введенные данные по-

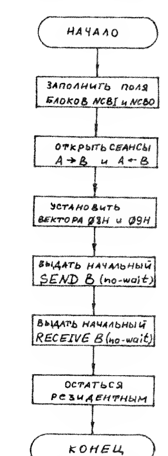


Рис. 3. Блок-схема инициирующей части утилиты CHPRL

[2]Извлекаемые данные при выполнении функций AH=0, 1, 10H, 11H прерывания INT 16H возвращаются в регистре AX (в AH — скан-код или расширенный ASCII-код, в AL — ASCII-код или 0).

мещает в резервное слово, расположенное непосредственно за клавиатурным буфером.

Как видно из вышеизложенного, для программ, запрашивающих ввод данных с клавиатуры через INT 16H (а их подавляющее большинство, включая и саму DOS), безразлично, введены данные с клавиатуры или просто занесены в клавиатурный буфер некоторой программой. Именно это обстоятельство и используется в CHPRL. Как уже говорилось, ее работа сводится к перехвату скан-кода и ASCII-кода, введенных с клавиатуры локального компьютера, пересылке их в удаленный компьютер и занесению в его клавиатурный буфер. Единственным местом перехвата скан-кода и ASCII-кода и их передачи в удаленный компьютер может быть обработчик прерываний от клавиатуры (вектор 9). Что касается приема скан-кодов и ASCII-кодов из удаленного компьютера, то на первый взгляд наиболее подходящее место для этих действий — обработчик прерываний по INT 16H. В ранних версиях утилиты CHPRL именно так и было, но, как вскоре выяснилось, что подходит для большинства программ, не приемлемо для Chess Master (для которой CHPRL и была задумана). Дело в том, что Chess Master для ввода данных с клавиатуры выдает непосредственно INT 16H только в самом начале работы, пока еще не сделан первый шахматный код. После того как он сделан, Chess Master, имеющий собственный обработчик прерываний от таймера, каждый раз, получив управление, проверяет, равна ли голова клавиатурного буфера его хвосту (пуст буфер или заполнен), и только в случае их неравенства выдает INT 16H. Интервалы времени между прерываниями от таймера Chess Master использует для анализа сложившейся шахматной позиции. Вот почему для приема данных из удаленного компьютера и их занесения в буфер клавиатуры в CHPRL используется обработчик прерываний от таймера (вектор 8). На рисунках 3-5 представлена упрощенная блок-схема основного алгоритма CHPRL. Давайте рассмотрим ее. CHPRL состоит из инициирующей части (рис. 3) и двух обработчиков прерываний: от таймера (рис. 4) и от клавиатуры (рис. 5)[3].

После запуска CHPRL инициирующая часть осуществляет следующие действия. Заполняет необходимые поля блоков NCBI и NCBO. Открывает сеансы A-»B и A«-B (здесь так же, как и в блок-схеме открытия сеансов на рис. 2, подставлены конкретные имена A и B). Сохраняет указатели на текущие обработчики прерываний от таймера (вектор 8) и клавиатуры (вектор 9) и устанавливает свои обработчики. Инициирует процесс обмена данными между двумя рабочими станциями, для чего выдает начальные команды SEND и RECEIVE в режиме no-wait (поле NCB_POST@ равно 0). Ввиду того, что к этому моменту еще нет данных для передачи, в качестве данных передается нулевое слово. И наконец, обеспечивает резидентность программы (AH=31H, INT 21H).

Все действия по обмену данными осуществляют обработчики прерываний. Прокомментируем их блок-схемы. Начнем с обработчика прерываний от таймера.

Получив управление, обработчик прерывания от таймера вызывает предыдущий обработчик (как обычно, путем PUSHF и CALL), благодаря чему обеспечивается нормальная обработка прерывания, включая выдачу приказа EOI (End Of Interruption) контроллеру прерываний. Затем, после сохранения регистров, проверяется, завершилась ли предыдущая команда RECEIVE, если да, то проверяется результат ее выполнения. Если команда выполнилась нормально, проверяется, какие приняты данные — начальные (нулевое слово) или нет. Если данные не начальные, проверяется, не принят ли сигнал к завершению от другой утилиты CHPRL (слово, содержащее FFFFH). Если принят сигнал к завершению, выполняется процедура завершения, включающая отмену незавершившейся команды SEND посредством команды CANCEL, закрытие сеансов с помощью команды HANGUP, удаление своего имени командой DELETE NAME, восстановление ранее сохраненных векторов и освобождение памяти, занимаемой CHPRL (AH=49H, INT 21H). Далее восстанавливаются регистры, и управление возвращается прерванному процессу DOS. Если же были приняты нормальные данные (скан-код и ASCII-код), они зано-

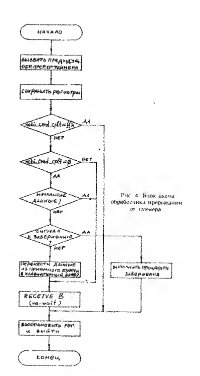


Рис. 4. Блок-схема обработчика прерываний от таймера

[3]На самом деле в состав CHPRL входит также обработчик INT 23H, необходимый для обработки Ctrl-Break, выданного пользователем в момент выполнения цикла установления сеансов. Этот обработчик отменяет незавершенные команды CALL и LISTEN посредством команды CANCEL, если один из сеансов успел открыться, закрывает его командой HANGUP и удаляет локальное имя командой DELETE NAME.

сятся в буфер клавиатуры. Затем выдается команда RECEIVE, запрашивающая прием очередных данных из удаленного компьютера. Если предыдущая команда RECEIVE завершилась ненормально, данные из приемного буфера не заносятся в клавиатурный буфер, но команда RECEIVE выдается. И наконец, если команда RECEIVE еще не завершилась, ни заносить данные в клавиатурный буфер, ни выдавать команду RECEIVE, разумеется, не надо. В любом случае, затем следует восстановить регистры и вернуть управление прерванному процессу DOS.

Обратимся теперь к блок-схеме обработчика прерываний от клавиатуры. Получив управление, так же как и рассмотренный выше обработчик прерываний от таймера, обработчик прерываний от клавиатуры в первую очередь вызывает предыдущий обработчик. Этим обеспечивается нормальная обработка прерывания и гарантируется попадание в клавиатурный буфер (локального компьютера) скан-кода и ASCII-кода, если они были сгенерированы. Затем сохраняются регистры и проверяется, завершилась ли предыдущая команда SEND. Если она еще не завершилась, регистры восстанавливаются и управление возвращается прерванному процессу DOS. В противном случае (SEND завершилась), осуществляется проверка, не ввел ли пользователь сочетание клавиш, запрашивающее завершение CHPRL (Alt-Shift). Это достигается путем проверки битов флажка клавиатуры по адресу 40H:17H, где клавиатурная BIOS отражает текущее состояние ряда служебных клавиш. Если завершение действительно затребовано пользователем, в буфер передачи заносится слово, содержащее FFFFH (сигнал к завершению партнера — CHPRL на удаленном компьютере), и выдается команда SEND. После этого выполняется процедура завершения, аналогичная описанной при рассмотрении обработчика прерываний от таймера. Далее восстанавливаются регистры, и управление передается прерванному процессу DOS. Если завершение не было запрошено пользователем, проверяется, имеются ли данные в клавиатурном буфере и если данные есть, они заносятся в буфер передачи, и выдается команда SEND, после чего управление получает прерванный процесс. Если нет, управление отдается ему сразу же.

Рис. 5. Блок-схема обработчика прерываний от клавиатуры

И в заключение несколько слов о конкретной реализации CHPRL. CHPRL — резидентная утилита, занимающая в памяти около 16 Кбайт (написана на MS C). Запускается она командной строкой вида: CHPRL «свое имя» «имя партнера». Как уже было сказано, завершается по Alt-Shift, введенного с одного из компьютеров, как на локальном, так и на удаленном компьютерах (пользователь удаленного компьютера получает при этом сообщение о причине завершения). Предусмотрена возможность включения/отключения запараллеливания клавиатур по вводу Ctrl-Shift. Помимо игры в шахматы утилита CHPRL может с успехом применяться для дублирования на удаленном компьютере действий, выполняемых пользователем на локальном компьютере.

В.Баевский

По материалам:

1. LANsmart Programmer's Reference, by D-Link Inc. — Datex Inc., 1989.
2. Brown Ralf, Interrupt list, Release 91.1, Last change 1/5/91.
3. IntrList version 1.02, by Window Book, Inc., 1990.
4. LAN Evaluation Report, 1986, by Novell Inc., Orem, Utah.
5. Mier Edwin, Controlling protocol confusing, перевод статьи в Computer World Moscow под названием "Как разобраться с протоколами", № 13, 1992.
6. Tech Help version 4.02, by Flambeux Software, 1990.



Шерлок Холмс против Джека Потрошителя

THE LOST FILES OF SHERLOCK HOLMES

Mythos Software /Electronic Arts. (c) 1992

Последняя четверть XIX века. Лондон. Лучшие силы полиции брошены на поиски убийцы, известного как Джек Потрошитель. Вводный мультфильм делает нас свидетелями убийства актрисы, и нам необходимо помочь Шерлоку Холмсу узнать, связано ли это дело с Джеком Потрошителем. Игра начинается, как и практически во всех литературных аналогах, на Бейкер Стрит, где Шерлок Холмс получает записку от инспектора Лестрейда из Скотланд Ярда с просьбой помочь в расследовании загадочного убийства. Напомним, что оба сыщика отличаются своим подходом к расследованию (знаменитый дедуктивный метод!), и обычно Лестрейд просит помощи у Холмса, только когда расследование заходит в тупик.

Итак, вместе с Шерлоком Холмсом и его бессменным помощником и другом доктором Ватсоном мы расследуем убийство актрисы.

Прежде чем перейти к рассмотрению самой игры, позволю себе небольшую цитату, которая позволит охарактеризовать художественную сторону игры:

"...Сыщик... странствует по Лондону, одинокий и свободный, как принц в волшебной сказке... Вечерние огни города начинают светиться, как глаза бессчетных домовых — хранителей тайны, пусть самой грубой, которая известна писателю, а читателю — нет. Каждый изгиб дороги, словно указующий перст, ведет к решению загадки; каждый фантастический контур дымовых труб на фоне неба, похоже, неистово и насмешливо намекает на значение тайны" (Г.К.Честертон. В защиту детективной литературы).

На мой взгляд, атмосфера Лондона передана на редкость удачно.

Как и в большинстве историй про Шерлока Холмса, начальное происшествие (убийство актрисы) является лишь только завязкой более сложного преступления. В ходе игры мы попадаем в ряд тупиковых ситуаций, прежде чем находим настоящего преступника. После того как первый подозреваемый помещен под стражу, возникает больше загадок, чем разгадок. Отметим, что литературная основа игры заслуживает внимания любителей творчества А. Конан-Дойла.

Интерфейс напоминает игры фирмы LicasArts: нижняя часть экрана содержит кнопки с названиями действий. Большинство предметов имеет предопределенные действия, что облегчает манипуляцию ими. Несмотря на простоту интерфейса, необходимо очень внимательно осматривать каждое место действия с тем, чтобы не пропустить важных улик.

Диалоги между игроком и оппонентом строятся с помощью списка. Обычно предлагается несколько фраз, из которых следует выбрать одну, необходимую в данном случае.

Перемещение по Лондону осуществляется по карте, причем новые места отображаются иконками, только если мы узнали о них в ходе расследования.

Интересной возможностью является журнал доктора Ватсона, в который заносятся все диалоги между Холмсом и другими персонажами. К окончанию игры мы получаем большой файл, который можно прочитать как увлекательную детективную историю. Содержимое этих записей может быть просмотрено также и в процессе игры, что может облегчить ее прохождение. К сожалению, в записки не заносятся действия, предпринимаемые Холмсом в ходе расследования.

Рассматриваемая нами игра — прекрасный пример реализации приключенческой игры, сопровождаемой интересным сюжетом и отлично выполненной графикой. В ходе расследования необходимо провести несколько химических опытов в домашней лаборатории Холмса.

Игра занимает 15 Мбайт дискового пространства и требует наличия видеоадаптера VGA. Наличие манипулятора "мышь" делает использование игры более удобным. Для стабильной работы требуется не менее 580 Кбайт свободной оперативной памяти.

А. Федоров

НОВОСТИ

Channel marketing назвал имена "входящих" и "выходящих" групп производителей IBM-совместимых компьютеров, подразумевая под этим названием торговые марки, которые предложены ("входящие") или сняты ("выходящие") с "каналов" рынка розничной продажи в этом году.

Доклад Channel о рынке розничной торговли "Кто входит, кто вышел и кто все еще рядом" называет производителей PC, вышедших на рынок в 1992 году, но ушедших с него; тех, кто вышел на него впервые; большое число тех, кто вышел на рынок, ушел с него и вернулся вновь.

Говорят, что такой суматохи на рынке розничной торговли PC никогда еще не было.

Например, компьютер марки ESP был сначала предложен фирмами CMS и Trigen, затем CMS ушла с рынка, а Trigen основала новую компанию с другим партнером, вновь предложив на рынке марку ESP. Также и компания Emerson ушла с рынка, но вернулась после того, как другая компания предложила ей деньги на лицензию.

Компании, ушедшие с рынка: Goldstar, Magnavox, KLH, Smith Corona, Emerson, Samsung, Sensor, Tandon, Wang, PC Brand, Everex.

Вышли на него с новыми изделиями: Canon, CMC, Leading Edge, EasyData, EiSys, Apple, Compaq, NEC, AT&T и Hewlett-Packard.

Все еще рядом: ESP, Laser, Partner и Emerson.

Некоторые компании сохранили свое положение на рынке. Это Packard Bell, Acros/Acer, AST и IBM. Также ожидается, что в 1993 году на рынок вступят компании Sony, Panasonic, Epson и Toshiba.

Newsbytes News, December 18, 1992

В каждой серебряной оправе есть пятно, как гласит закон Мэрфи, и, кажется, фирма **Intel** входит в темный период своей жизни. Несмотря на рекордный доход и подъем на вершину в полупроводниковой промышленности, акция FTC (Федеральной Арбитражной Комиссии) против микропроцессорного гиганта может быть уже на пороге.

Доход Intel в 1992 году достиг высшей отметки в 5,84 миллиарда долларов США, а годовой прирост дохода составил 1,07 миллиарда долларов США, что на 22 и 30 процентов, соответственно, больше, чем в 1991 году.

В четвертом квартале 1992 года было продано компьютеров на 54% больше, чем в том же квартале 1991 года, что составило 1,86 миллиардов долларов США. Президент компании Intel Эндрю Гроув (Andrew Grove) сообщил о том, что компания ставила своей целью выпуск 4-5 процессоров второго поколения (Intel 486 и Intel 386), но реальный выпуск превысил это число. Компания Intel также считает успехом 1992 года одновременное признание компаниями Apple, IBM и Microsoft программного обеспечения к видео-системе Indeo, разработанного ею; начало производства микропроцессоров 486 DX2 и признание компанией Instat семейства процессора Intel 960 процессором с самым совершенным набором инструкций среди RISC-процессоров.

Компания сообщила, что доход на каждого сотрудника составил 232.000 долларов США и они также получили трехнедельную дополнительную оплату из призового фонда компании, который в 1992 году составил 61 миллион долларов.

Однако после прихода к власти администрации Клинтона многие поверили в то, что FTC может быстро свернуть свое двухлетнее расследование деятельности Intel. Компании Cyrix и Advanced Micro Devices имеют незаконченные антимонопольные тяжбы в отношении микропроцессорного гиганта, и FTC сейчас активно контактирует со своими юристами по поводу этого расследования.

Недовольство высказывается в связи с тем, что компания Intel использует свое положение монополиста, чтобы помешать другим компаниям использовать микропроцессоры, отличные от разработанных Intel. Компании, покупающие оборудование у Intel, имеют ограничения, определенные Intel'ом, то есть могут купить только ограниченное число чипов и комплектующих.

Компания — производитель оригинального оборудования Acer обратилась за помощью в компанию Cyrix, поскольку Intel грозится ужесточить ограничения для его компании, так как она чаще использует процессоры Cyrix, а не Intel.

Статья в "San Francisco Examiner" рассказывает о том, что Intel также посылает письма в компании, производящие компьютеры без микропроцессоров Intel, с просьбой заплатить за каждый компьютер, выпущенный для использования ее патента Крауфорда (Crawford), что называется "связывание патентом".

Как известно, никто еще не заплатил Intel за использование патента, который определяет, как работает микропроцессор с памятью, когда используется многозадачное программное обеспечение типа Windows, OS/2 или Unix, но требование Intel об оплате не является законным.

"Во-первых, плата требуется с тех, кто использует не Intel-процессоры, тогда как OEM (оригинальные разработчики аппаратуры) на базе процессоров Intel свободно используют запатентованную технологию, и, во-вторых, судья в этом деле постановил, что все патенты Intel покрываются лицензиями Cyrix от Intel.

Компания Advanced Micro Devices (AMD) также уверена в том, что Intel занимается незаконной практикой, и предоставила по запросу FTC информацию о своем арбитражном разбирательстве с Intel. Как говорят, предполагалось, что оно закончится за шесть недель, но длится дольше, чем вторая мировая война.

То, что делает FTC, — тайна. Она даже не объявила официально о том, что ведет расследование деятельности Intel, однако все знают об этом. Предположения, интенсивно исходящие от FTC, не будут иметь никакого успеха, FTC может лишь наложить штраф, вести юридическую деятельность или заставить Intel поделиться технологиями.

Newsbytes News, January 13, 1993

Dell Computer Corporation UK запустила в производство 18 новых машин на базе процессора Intel 80486, аналогичных компьютерам, выпуск которых начат в Америке перед Рождеством.

Новые персональные компьютеры, к выпуску которых можно приступить сразу же, базируются на Intel 486SX, DX и DX2 наборах чипов, работающих на частотах 25, 33 и 50 МГц. Они делятся на четыре семейства:

Серия 486/L — пять низкоуровневых персональных компьютеров, осно-

ваиных иа архитектуре ISA по цене от 1179 фунтов стерлингов.

Серия 486/M — пять систем с архитектурой ISA среднего размера по цене от 1279 фунтов стерлингов.

Серия 486/ME — еще пять машин с архитектурой EISA, стоимость от 1769 фунтов стерлингов.

486/T — новое семейство машин Dell, состоящих из трех напольных компьютеров с архитектурой ISA, начиная с 486 DX систем (по цене от 1739 фунтов стерлингов).

Цены на эти машины включают в себя "базовую" спецификацию, состоящую из 4 Мбайт памяти (8 Мбайт на 486/ME и 486/T), жесткого диска объемом 120 Мбайт, SVGA-монитора, мыши, DOS и Windows.

Компания уже заменила основной поток выпуска 486 машин на свои новейшие системы. Их работа более производительна и обладает качествами, более удобными для пользователей.

Все четыре системы имеют ряд новшеств, которые улучшают работу и облегчают их использоване, включая встроенную видеоподдержку с локальной шиной, повышающую производительность при работе с Microsoft Windows и другими графическими программами.

333s/l — это компьютер на базе процессора 386SX с частотой 33 МГц, предназначенный для домашнего использования или для пользователей, впервые купивших компьютер. Цены — от 859 фунтов стерлингов, с базовой памятью 4 Мбайта и жестким диском 80 Мбайт.

Создатели системы говорят, что она разрабатывалась специально для неопытных пользователей и домашнего пользования, то есть для тех, кто еще не может заплатить немного больше, чтобы купить 486 компьютер, но хочет иметь мощную машиу.

Из всех новых машин 320SLI, возможно, самая впечатляющая. Этот ноутбук, весящий меиее 3,6 фунтов (1.6 кг), работает на микропроцессоре 80386 на 20 МГц. Базовая конфигурация состоит из 4 Мбайт памяти, жесткого диска 80 Мбайт, мыши Ballpoint, DOS и Windows (все вместе — 1274 фунтов стерлингов). Дополнительно может поставляться память до 10 Мбайт и жесткий диск 120 Мбайт.

Многих пользователей привлекает идея миниатюрных персональных компьютеров, но они в ней быстро разочаровываются из-за многих ограничений. Но 320SLI — это изделие, о котором многие мечтали. В то же время оно отвечает нуждам многих пользователей стандартных ноутбуков.

Одной из основных особенностей 320SLI является корпус размером на 40% меньше, чем у конкурентов. Его объем был уменьшен за счет использования нового супертоикого жидкокристаллического дисплея, который потребляет на 75% меньше эиергии. Это, в свою очередь, уменьшает энергопотребление машины и требования к батарее. Жидкокристаллический дисплей также может быть устаиовлен прямо по месту, что также уменьшает объем нзделия.

Newsbytes News, January 11, 1993

Президент компании **Zenith Data Systems** (ZDS) вынужден был уйти со своего поста, так как потерял поддержку в компании, расположеиной в Чикаго, Иллииойс.

Компания ZDS потеряла осиовной пентагоновский контракт в США. Работа, проводимая с американскими ВВС, прекратилась, когда стало ясно, что правительство не выделит на нее ассигиований.

Кроме того, ZDS понесла серьезные потери и независимо от этого контракта. Ее основатель, компания Bull, SA (Париж) получила бы в прошлом году больший доход, если бы не трудности, испытанные ею в США.

ZDS ие смогла остаиовить поток неудач, так как контракты, которые оиа ожидала как в США, так и в Европе, до сих пор ие поступали.

Компания также сильно пострадала от конкуренции с такими компаниями, как Dell, Compuadd и Gateway.

Пост президента вместо Песатори займет Джэквес Ноэл (Jacques Noel), многие годы работавший на Bull.

Newsbytes News, January 13, 1993

Компания **Dell Computer** открывает новые филиалы в Японии и Австралии, так как считает это частью своей кампании по расширению своего участия в мировом рынке компьютеров.

Компания объявила, что филиалы предложат покупателям полный набор обычных компьютеров, серверов и портативных компьютеров.

Как сообщалось ранее, цены иа системы на базе 386 процессора (33 МГц) в Японии составят от 800 долларов. Компания будет поставлять совместио с компьютером прикладные программы и сетевое программиое обеспечение, гараитировать установку оборудования у заказчика на следующий день после приобретения и неограниченную техническую поддержку.

Теперь, когда японская экоиомика борется с трудностями, потребители будут лучше осознавать ценность товара.

Компания встретила в Японии сильную конкуренцию. NEC имеет крепкое положение, IBM и Compaq также имеют свои представительства. Конкуренция очень велика, все компании участвуют в войне цен. Dell намерена быть конкурентоспособной, но нигде, ни в США, ни где-либо еще она не будет устанавливать очеиь низкие цены. Компания уверена, что ее продукция выгодно отличается от конкурирующих товаров. В то же время Dell может производить более дешевую продукцию для рынка, так как обладает эффективными методами производства.

Австралийский рынок — второй из самых больших рынков в азиатско-тихоокеанском регионе. Он очень сложиый и конкурентоспособный.

Японский рынок готов принять Dell, которая выходит на него в то время, когда Япония переживает относнтельио медленный экономический подъем.

Дела фирмы в Японии будет вести Генеральный меиеджер Катсуми Тизука (Katsumi Tizuka), который был главой Дальиевосточного отделения Dell с 1988 года. Компания будет продавать PC непосредственно потребителям и через три основные японские торговые компьютерные фнрмы.

Newsbytes News, January 21, 1993

Фирма ИнФоС, создатель Интегратора "Виктория", выпустила **библиотеку Victory Tools** для разработчиков, которая имеет следующие отличительные особенности:

- позволяет создавать программы сразу для двух операционных сред — DOS и Windows. Изначально прикладиая программа создается в идеологии Windows, с использоваиием функций Windows API, а затем, по желанию разработчика, может быть оттранслирована для любой из двух поддерживаемых сред;
- при использоваиии в среде DOS программа может работать как в графическом, так и в текстовом режимах, сохраняя при этом основиые элементы интерфейса Windows. Преимуществом такого подхода является возможность эффективного выполнения Windows-подобных программ на компьютерах с процессорами 80286 и ниже.

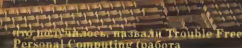
Библиотека поставляется в комплекте со средой для выполнения программ, которая обеспечивает режим переключения задач, созданных на основе этой библиотеки. Те же самые задачи после перекомпоновки (не изменяя исходного текста) могут выполияться в среде Windows.

Желающие озиакомиться с данной библиотекой и технологией создання программ могут посетить стенд фирмы MicroMax (№ 66) иа выставке Комтек'93.

КомпьютерПресс

Производительность. Качество. Надежность. По разумным ценам.





Мы знаем, что скорость, качество и надежность персональных компьютеров всегда были важны для Вас. Но мы знаем и то, что сейчас для Вас важна еще одна вещь: компьютер должен работать, не преподнося Вам дополнительных проблем.
Поэтому в компьютеры HP Vectra было добавлено новое измерение, и то, что получилось, назвали Trouble Free Personal Computing (работа на персональном компьютере без проблем). Это комплекс возможностей, позволяющих нашим компьютерам превзойти Ваши ожидания.
Беспрецидентная надежность, исключительная эргономичность, простая установка, удобность в эксплуатации и обслуживании.
И, кроме того, встроенная поддержка сети, средства обеспечения безопасности и хорошие возможности наращивания.
Это делает новое семейство Vectra способным удовлетворить потребности завтрашнего дня, так же как и сегодняшнего.

113035 Москва, ул.Осипенко 15, корп.2, офис 207.
Телефоны: (095) 237-66-81, 230-56-12, 220-27-59
Факс: (095) 230-21-82





